ROADBLOCKS TO SOFTWARE MODERNIZATION

THESIS

Dyan E. McClamma, Captain, USAF

AFIT/GRD/ENV/O6M-09

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

AFIT/GRD/ENV/O6M-09

ROADBLOCKS TO SOFTWARE MODERNIZATION

THESIS

Presented to the Faculty

Department of Engineering and Management

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Research and Development Management

Dyan E. McClamma, BS

Captain, USAF

March 2006

AFIT/GRD/ENV/06M-09

ROADBLOCKS TO SOFTWARE MODERNIZATION

Dyan E. McClamma, BS

Captain, USAF

Approved:

_________________________________ __________
Michael T. Rehg, PhD, (Chairman) Date

_________________________________ __________
Timothy J. Halloran, Lt Col, USAF (Member) Date

_________________________________ __________
Ross T. McNutt, Lt Col, USAF (Member) Date

AFIT/GRD/ENV/06M-09

Abstract

Failed or troubled modernization efforts, such as the multi-million dollar 1997–2000 ROCC/SOCC failure, are a serious acquisition problem for the Air Force. Using both historical data and a survey of current Air Force software acquisition program key staff, this research examined the Air Forces ability to modernize legacy software systems.

The search of historical program data, to identify trends or similarities between known failed software modernization efforts, failed to uncover sufficient data for analysis. This lack of project data indicates a knowledge management issue (i.e. lessons learned are not recorded and stored so that they can be accessed by other programs) in the acquisition community.

The Phase II survey gathered data on current software programs and addressed the recommendations of the 2000 Defense Science Board (DSB) Study on Software. The goal was to determine first, had the recommendations been implemented, second, did program characteristics effect implementation, and third, did implementing the recommendations lead to program success. The survey results indicate that most of the recommendations of the DSB are not in practice in the acquisition community. They also indicate that support programs are more likely to have implemented the recommendations than are weapons systems.

Acknowledgments

My sincerest appreciation goes to my faculty advisor, Lt Col Timothy J. Halloran, for his guidance and support throughout the course of this thesis effort. His insight and experience were crucial to my success. I also thank Dr. Michael Rehg for his support most especially throughout the survey construction, administration, and analysis for this thesis; his experience and expertise were invaluable. I would be remiss to leave out my husband and children whose support and understanding made my successful completion possible and worthwhile.

Dyan E. McClamma

Table of Contents

Page

Page

List of Figures

Page

Page

List of Tables

Page

ROADBLOCKS TO SOFTWARE MODERNIZATION

I. Background

In March of 1997, Litton Data Systems was awarded a $58,003,369.00 cost plus award fee contract for the development of a modernization system to replace the Region/Sector Air Operations Centers legacy systems (ROCC/SOCC). (GlobalSecurity, 2005) The ROCC/SOCC systems were initially declared fully mission operable in the early 1980's, so by the time this contract was let the systems were nearly 20 years old, and had undergone many years of updates and modifications. By 2000, Litton had failed on the modernization contract with millions of dollars spent and nothing to show for it. In 2004 the original Hughes mainframes were still running at the centers, now more than 20 years old and a maintenance nightmare. This system is not the only one in the Air Force inventory to experience this type of modernization issue - there are many others. The Standard Base Supply System (SBSS), the Defense Travel System (DTS), and the Military Personnel Database System (MilPDS) are all software systems that experienced costly modernization problems.

The failed or troubled modernization of systems like these were the motivators for this research. Is it possible to identify commonalities among these troubled modernization efforts in order to prevent them in the future? Knowledge of what drives a system to become a modernization nightmare could enable us to ensure future systems are upgradeable.

Software is a crucial component to military technology systems and as such is a critical high-risk element to nearly all, if not all, major weapon systems and support systems. The modernization of these systems to current technological capabilities is seemingly hampered by the technicality and high risk of their software components. Modernization in this case is defined as the creation of a system performing the same function, often with enhancements, on a new platform with modern software languages and engineering practices. Several studies such as those conducted by the Defense Science Board Task Force and the General Accounting Office report that the greatest difficulties with software-intensive programs are generally not with the technicality but with the inability of program managers to manage software efforts.

Numerous studies on new software development have reported poor management of software and a lack of technical experts as the reasons for the demise of new developments. The November 2000 Defense Science Board (DSB) task force reported that of 134 previous recommendations by science panels, Defense Science Boards, and the National Research Council, only 18 are in policy and only 3 in practice despite the fact that all were still deemed valid recommendations (DSB, 2000). However, there has been no research that we are aware of to determine if the same or similar factors are what cause the failure of software modernization efforts.

The goal of this research is to uncover potential reasons why the software for many major support and weapon systems have proven difficult or impossible to modernize despite the fact that they have lived far beyond their intended system lifecycle.

These systems were developed initially with less capable technology but for reasons to be discovered have been difficult or impossible to modernize.

Problem Statement and Research Questions

This research presents results of a study focused on three areas; first, discovering if there are commonalities among troubled or failed software modernization efforts, second, discovering if the Air Force acquisition community has adopted the recommendations of previous defense software studies, and third, to identify indicators that implementation of the previous recommendations affected the management success of the software and software intensive programs. The primary area of interest is software modernization, which is defined in this case as updating existing software systems to enable the use of modern hardware and fully leverage advances in technology. There has been research accomplished on new software development with similar findings that continue to report similar recommendations, yet very few of the recommendations have been acted upon by the Air Force.

This research investigates the following questions:

1. Has the Air Force implemented any of the past recommendations made by DOD agency task forces on software?

2. Do the characteristics of a system affect whether software acquisition management recommendations for improvement are implemented?

3. If the recommendations to improve software acquisition management made by the DSB were implemented, did they positively affect the outcome of the program?

Methodology

To gather data on past failed or troubled modernization efforts a search was conducted via personal contacts, involved agencies and known repositories of information such as the defense technical information center (DTIC). The search was approached from the acquisition, contracting, and cost sides of the programs to locate any type of historical data that may indicate when issues were first identified and what type of problems caused the demise of these systems.

To gather insight into implementation levels of the 2005 DSB recommendations a survey was developed and administered to software acquisition personnel. The population of interest was the system program directors (SPM), deputy program group managers (PGM), system support managers (SSM), development system managers (DSM), and key staff of the software acquisition program. Programs of interest were programs that were terminated by the MDA prior to deployment. The Joint Ammunition Management Standard System (JAMSS), the Defense Joint Accounting System (DJAS), the Defense Procurement Payment System (DPPS), the Region & Sector Air Operations Center Modernization (ROCC/SOCC), and the Standard Base Supply System (SBSS) were initially identified as systems of interest. The expected sample size of the survey were the key personnel of the above listed systems in addition to key personnel of all other AF software and software intensive systems identified during the course of research.

The constructs measured included program management's software knowledge and experience, the use of software engineering processes, and team member's familiarity with them, how software contractors were chosen, and the likelihood that metrics were used to guide management decisions.

Implications

This research identified that there is a knowledge management issue in the software acquisition community. By knowledge management we mean, the collection, organization, and storage of knowledge and experiences of individuals and groups in an organization along with ensuring this information is available for others to benefit from it. While the initial area of interest is software modernization it will be shown in later chapters that there is a significant lack of data available to analyze similarities or causes of past modernization failures. Lessons learned are extremely difficult to uncover on failed programs. Learning from mistakes is an important part of improving our abilities to manage software and is an issue that needs further attention. We also identified that some programs have implemented the recommendations of the 2005 DSB for software management however, this study did not uncover any evidence that the implementation of these recommendations had a positive affect on the successful management of the programs. This is another area that would benefit from further study to determine if over time the programs that have implemented the software management process recommended by the DSB are more successful than those that have not. It is not possible to say that a program is successful by looking at a single moment in its existence, however these indicators watched over a period of time could show trends in what makes a successful program successful and an unsuccessful program unsuccessful.

Outline

The following chapters detail the research and results. Chapter 2 summarizes past research and describes how it is used as a basis for the research questions. Chapter 3 discusses the methodology used for gathering and analyzing data. Chapter 4 presents results and analysis of data. Chapter 5 provides the conclusions and recommendations based on these results.

II. Literature Review

This chapter summarizes the findings of previous research to give the reader an understanding of the problem, the research accomplished to date, and areas in need of further investigation. The primary documents discussed are findings by the Defense Science Board (DSB) and the General Accountability Office (GAO). These two entities have conducted the majority of the research in the area of Air Force software acquisition management. No research was found that specifically addresses Air Force software modernization. The methodologies used and findings of these past studies are discussed to build the basis upon which the constructs of this research were developed.

Description

Overview

From its inception, the Air Force has acquired and managed software. While the Air Forces software acquisition processes have improved over time, it is our belief that much more can be done to increase the efficiency of our software acquisition abilities. Software engineering, and therefore the development and management of software, is a relatively young field. It does not have the benefit of decades of research and improvements as do fields like mechanical and civil engineering.

"A Government that works better and costs less requires efficient and effective information systems." (Federal Register, 1996) This is the leading quote into a discussion of the adoption of the Paperwork Reduction Act of 1996 and the Information Technology Management Reform act of 1996 indicating that at that time the DOD, to include the Air Force, was aware that things were in need of a change. A vast improvement in the way information technology systems were being acquired and managed was needed and being pushed from the highest level. Prior to the adoption of these reports many of the Air Forces support systems were already becoming dinosaurs. For example, the Regional Operational Control Centers (designator AN/FYQ 93) were running code that was developed in the 70's and was running on Hughes 1970's technology mainframes (source?). The Standard Base Supply System was running on a 1970's technology Unisys mainframe with primarily COBOL code. Currently, the Defense Finance and Accounting systems are decades old along with the Defense Travel System. Missile warning satellites are nearing the end of their useful life (GAO, 2004). The list goes on, and it is clear the DOD and the Air Force have a need to improve acquisition and management of information technology systems to modernize these antiquated information systems. The need for modernization is increased as old hardware becomes un-maintainable. The parts are no longer manufactured, or if they are manufactured it is at a high cost to the government as the solitary consumer. Cornella-Dorda et. al. (2000), captured the reality of the situation well in their introduction to "A Survey of Legacy System Modernization Approaches" with the following:

> *"Information Systems are critical assets for modern enterprises and incorporate key knowledge acquired over the life of an organization. Although these systems must be updated continuously to reflect evolving business practices, repeated modification has a cumulative effect on system complexity, and the rapid evolution of technology quickly renders existing technologies obsolete. Eventually, the existing information systems become too fragile to modify and too important to discard."*

In the case of the ROCC/SOCC modernization the primary cause of modernization failure was the inability to produce the software for the modernization effort. The Litton contract for "ROCC/SOCC modernization" was cancelled in the mid '90s by the Major Decision Authority (MDA) with $200 million spent on development and nothing to show for it. Cases such as this are what prompted the investigations by the Defense Science Board and General Accountability Office as described below.

Prior Work

Software studies conducted since 1981 within the DOD and related to Air Force software are listed in table 1. As the list shows the amount of research done in this area is not significant and a majority of the research was accomplished in the late 80's to early 90's. It is interesting that each of these studies reiterate many of the findings of the previous studies. It appears that the past recommendations either are not being implemented, are not possible to implement, or aren't perceived as valid recommendations by the acquisition community. This research attempts to answer three questions; (1) are the past recommendations being implemented (2) do program characteristics affect whether a program has implemented the past recommendations, and (3) has implementation improved the success in acquisition of software/software intensive programs?

Table 1 - Previous government sponsored studies on software

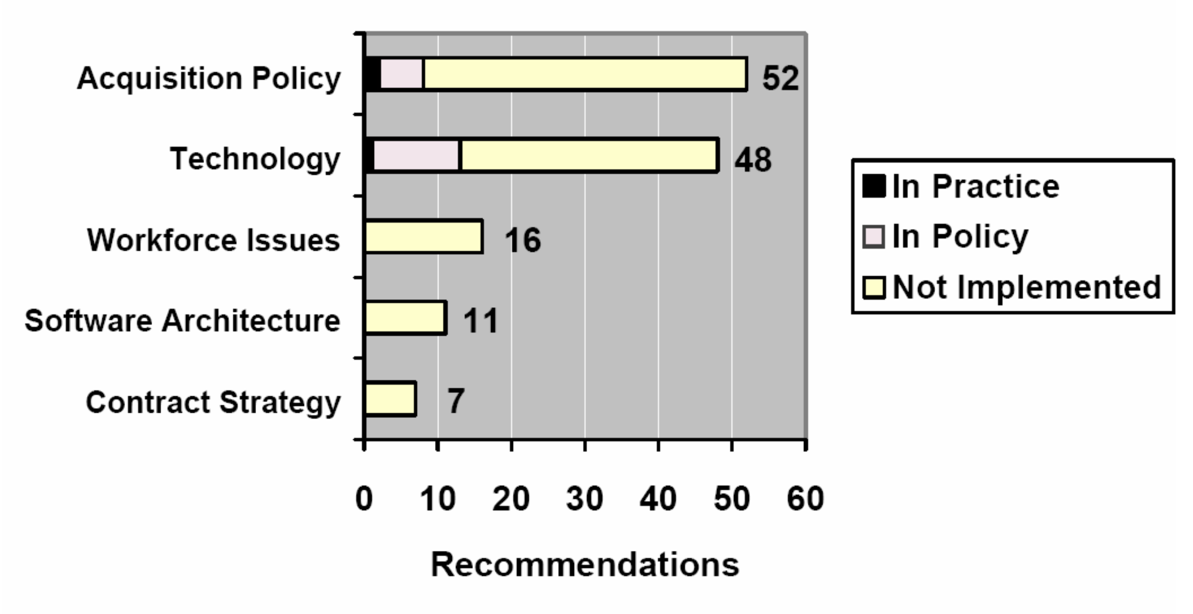
Previous Studies	Year
DSB Summer study on Technology Base	1981
Joint Service Task Force on Software Problems	1982
AF SAB High Cost and Risk of Mission Critical Software	1983
CODSIA Report on DoD management of Mission-Critical Computer Resources	1984
DSB Task Force on Military Software	1987
Ada Board Response to DSB Task Force	1988
Summer Report on Defense-wide Audit of Support for Tactical Software	1988
Workshop on executive Software Issues	1988
Army Materiel Command Study	1989
Software Technology Development and Deployment Plan for DoD Technology Base	1989
Adapting Software Development Policies to Modern Technology	1989
The Report of the AMC Software Task Force	1989
Scaling Up: A Research Agenda for Software Engineering - computer Science and Technology Board Research Council	1989
Draft DoD Software Master Plan	1990
Draft DoD Software Technology Strategy	1991
AF SAB Study on Information Architecture	1993
Study on Military Standards Impacts on the Acquisition Process	1993
Draft Software Action Plan working Group Report	1993
Evolutionary Acquisition Study, AFCEA	1993
U.S General Accounting Office, DoD Information Technology: Software and Systems Process Improvement Programs Vary in Use of Best Practices	2001
General Accounting Office Report to the committee on Armed Services, US Senate. Defense Acquisitions - Stronger Management Practices are Needed to Improve DoD's Software Intensive Weapon Acquisitions	2004

Defense Science Board Study

Beginning in September of 1999 and concluding in November 2000 the Defense Science Board task force on Defense Software led an investigation into the management and acquisition of DOD software. The objectives of the task force were as follows:

Review all findings of previous DoD wide studies on software development and acquisition; identify any changes in the current software development and acquisition practices since previous studies; assess the current state of both DOD and Commercial software development; and make focused recommendations to improve the performance on DOD software intensive programs.

The task force reviewed the reports of six major DoD-wide studies dating from 1987. The task force found that from these studies 134 recommendations were made however very few of the recommendations had been implemented into practice in Air Force software development and acquisition. Figure 1 depicts the categories of recommendations and indicates the numbers of each that were made and the numbers implemented. Not only did they find that the recommendations had not been acted upon they deemed them all still valid recommendations. This was the major finding of the board "the DOD's failure to implement these recommendations is most disturbing….Clearly there are inhibitors within the DOD to adopting the recommended changes."(DSB, 2000) Within today's acquisition programs it would be difficult to find a program that does not contain a software component. The DSB points out that "software is rapidly becoming a significant, if not the most significant, portion of DOD acquisitions." (DSB, 2000) I would venture to say that software components are the most significant portion of DOD acquisition; it is a major driver in schedule and cost for the majority of support and weapons systems today. As seen in figure 2 the trend was clearly moving in that direction in 1995. Eleven years later the Air Force, as the rest of society, has become even more computer dependent. Software acquisition efficiency is key to maintaining our status as a world military leader.



Source: Report of the Defense Science Board on Defense Software Nov 2000

Figure 1 - Categories and status of prior recommendations

The 2000 DSB task force's evaluation relied completely on inputs from "a representative sampling of programs and new technology efforts" - there was no detailed quantitative analysis or evaluation of individual topics. The focus of the task force was to provide a small number of recommendations that could be implemented quickly.

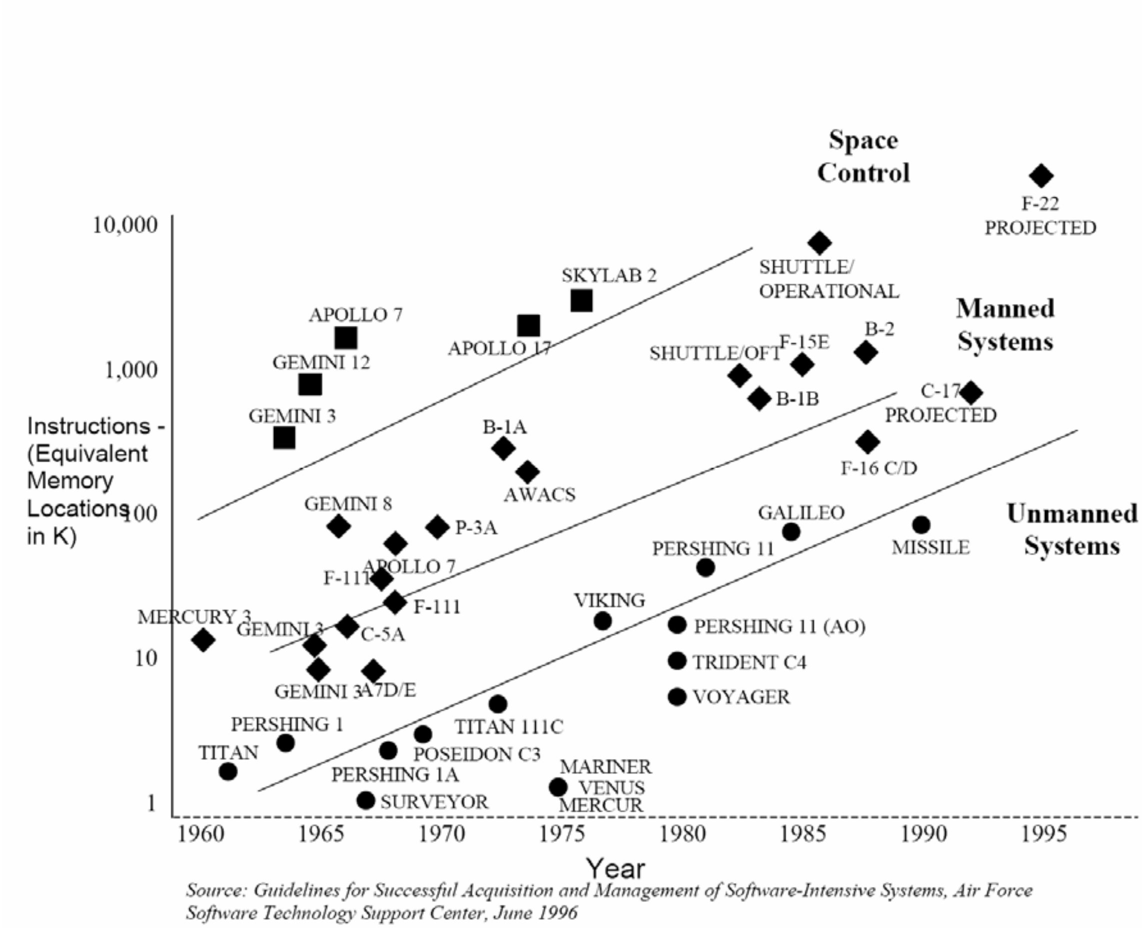


Figure 2 – Code size/complexity growth of software in Air Force acquisitions

"Many previous studies have provided an abundance of valid conclusions and detailed recommendations. Most remain unimplemented. If the military software problem is real it is not perceived as urgent. We do not attempt to prove that it is; we do recommend how to attack it if one wants to" (DSB, 1987) This quote was also quoted by the 2000 DSB on Defense Software, stating that sadly despite the 13 years between the two studies the findings were "strikingly similar." (DSB, 2000)

The 2000 DSB published in their report six recommendations:

1. “Stress [software] past performance and process maturity.: (DSB, 2000:ES-2) In other words, choose contractors with a proven record of accomplishment. A major government program is not the place to make a mark and prove ability, it is a place to put to use skills learned through experience.

2. “Initiate independent expert reviews.”(DSB, 2000:ES-3) These reviews are intended to ensure programs are being properly executed in cost, schedule, and performance. These reviews also enable the sharing of technical expert knowledge across programs.

3. “Improve Software skills of acquisition and program management.”(DSB, 2000:ES-3) The board found that DoD acquisition personnel were not adequately trained on software intensive program acquisition causing mismanagement of software programs.

4. “Collect, disseminate, and employ best practices.”(DSB, 2000:ES-3) The DoD needs to learn from past mistakes and educate the field on pitfalls to avoid in order to become successful software acquisitionists.

5. “Restructure contract incentives.”(DSB,2000:ES-3) Employ commercial performance incentive practices. The task force found the largest difference between commercial and DoD was the performance incentives. “In the DoD environment, profits are typically limited to 15% with little penalty for performance failures. In the commercial, market profits of 30% are common and poor performance can quickly lead to termination with significant financial liabilities.” (DSB, 2000:19)

6. “Strengthen the technology base.”(DSB,2000:ES-4) The DoD needs to maintain key researchers in order to produce technology not provided by the commercial marketplace.

At the same time the DoD needs to leverage commercial technology, there is no need for duplication.

"In December 2002 Congress required the Secretaries of each military service and the head of each defense agency that manage software-intensive acquisition programs to develop process improvement programs for software acquisition." (GAO, 2004:2) Based on this direction Congress subsequently requested that the GAO evaluate the performance of the services to meet these objectives.

In March of 2004 the United States General Accounting Office (GAO) provided a report to the Committee on Armed Services, U. S. Senate entitled "Defense Acquisitions: Stronger Management Practices are Needed to Improve DOD's Software-Intensive Weapon Acquisitions." Their purpose was to "identify the practices used by leading companies to acquire software and to analyze the causes of poor outcomes of selected DOD programs." and "evaluate DOD's efforts to develop programs for improving software acquisition processes and to assess how those efforts compare with leading companies." (GAO, 2004:)

Similar to the findings of the DSB the GAO's main findings were that "software acquisition outcomes were poor for programs that did not use an evolutionary approach, disciplined processes, and meaningful metrics." (GAO, 2004) The GAO study looked at five major DOD software intensive weapon system acquisitions; Tomahawk, F/A-18 C/D, F/A-22, SIBRS, and Comanche. They found mixed results among these programs. The conclusion was that when programs had a smaller evolutionary product with manageable requirements, used disciplined development processes with reviews, and collected and used metrics in decision making they delivered with less cost increase, and less schedule slippage.

The GAO also looked at five commercial companies via a literature search and structured interviews. Those companies were Computer Sciences Corporation (CSC), Diebold, Incorporated, General Motors, Motorola GSG, and NCR. From these companies the major lesson taken away was "The right environment reduces software development risk." The right environment is defined as one that focuses on evolutionary product development, adheres to well-defined, understood, and realistic requirements, and encourages continuous process improvement. This is consistent with the findings of the DoD software programs reviewed; those with this "right environment" were in the end more successful.

The GAO found that the services, to include the Air Force "have created a more conducive environment for software acquisition and development." The environment they speak of for the Air Force is a baseline of practices and suggested courses of action, at the time of the GAO study these recommendations were still being planned for; they were not in place for all Air Force software acquisition programs.

Data Search

In order to locate historical information about programs of interest it was first important to identify readily available sources or repositories of information. The largest and first repository to check is the Defense Technical Information Center (DTIC). DTIC is a DoD Field Activity under the Under Secretary of Defense for Acquisition, Technology and Logistics, reporting to the Director, Defense Research & Engineering

(DDR&E). A primary mission of DTIC is to "Provide centralized operation of DoD services for the acquisition, storage, retrieval, and dissemination of Scientific and Technical Information (STI) to support DoD research, development, engineering and studies programs."

"DoD-funded researchers are required to search DTIC's collections of technical reports and summaries of ongoing research to ensure unnecessary research is not undertaken" (DTIC web page.) Not only are researchers required to search the collection but DoD components are responsible, per DOD Directive 3200.12, for ensuring that DTIC is provided with all pertinent material resulting from Research, Development, Test, and Evaluation (RDT&E) programs (DTIC web page.) While the direction is there for DTIC to be a key repository for research and development information sharing, DTIC lacks an enforcement ability to guarantee compliance. While DTIC provides a valuable service and contains literally hundreds of thousands of documents, the lack of enforceability leads to a less then desirable outcome. There is much valuable information that is not captured via DTIC.

Some other government sources for technical information sharing are the Knowledge Web Services and Contractor Performance Assessment Reporting System (CPARS). Knowledge Web Services (KnWS) is managed by Techolote Inc. for the Air Force (Le Blanc, 2004.) KnWS provides a Web-based application to manage information of projects for organizations. KnWS is intended to provide wide spread distribution, retrieval, and storage of information, including: cost estimating relationships (CERs), documents, models, and references (Le Blanc, 2004.) "CPARS is a web-enabled application that collects and manages a library of automated CPARS." (CPARS website, 2005) A CPAR is an assessment of a contractors performance both positive and negative on a specific contract for a specific period of time (CPARS website, 2005.) CPARS is managed by the Naval Sea Systems Command.

A thorough search of each of these sources and more was completed during this research effort. It will be shown in later chapters that the data needed to learn from our past mistakes is not being saved in a manner to facilitate widespread information sharing and therefore hinders rapid process improvement.

Summary

The defense science board indicated that many of the troubled programs reviewed had readily identifiable fundamental problems such as a lack of disciplined program management or software development processes. Cost, schedule, and requirement baselines were unrealistic or did not exist making it impossible to track the programs progress. Contractor teams chosen did not have the proven skills to complete the program. Each of these issues is extremely similar to the findings of earlier research done by the DSB and the GAO. This research takes a deeper look into these claims by conducting a more intense data search and conducting a survey of personnel currently working programs in the field. The goal is to investigate the thought by the DSB that past recommendations are not being followed and to show that if they are in fact being followed, is there an effect on the successful outcome of the program.

III. Methodology

Chapter Overview

The purpose of this chapter is to describe the method used to test the research questions defined in Chapter 1. First, a background investigation was completed to uncover lessons learned from past failed or troubled software modernization efforts. The goal was to analyze the data from several different programs and determine if there is a common theme or themes causing a breakdown in Air Force software modernization practices. Second, a survey was developed to measure whether or not program teams follow the recommendations of the Defense Science Board (DSB) and the General Accountability Office (GAO) as published in their respective studies. Test subjects were selected based on their position in a program office of a software or software intensive system. The chapter concludes with a discussion of the statistical methods used to analyze the data gathered.

Research Design

The DSB and GAO studies discussed in chapter 2 are broad overviews of the state of AF software acquisition. This research was designed to take a deeper look into what is currently happening in AF software acquisition practices. The previous studies were mainly reports on a collection of briefings given to the board. The 2000 DSB members were provided 46 briefings on troubled programs, industry trends, and AF practices. The report does not indicate that more detailed research was done other than to synthesize these reports into a projection of what was causing trouble in AF software acquisition. To take a deeper look, data was needed to show first, has the AF software acquisition community made changes per the DSB's recommendations and second, if changes were made did those changes have an impact on the success of the program or its anticipated success for those still in early lifecycle stages. This study was conducted in two phases:

- Phase I - A search for any pertinent historical data was accomplished in an effort to develop a picture of what was happening in software acquisition
- Phase II - A cross-sectional between-cases survey was developed and administered to gather data from the field on changes implemented as recommended by past studies and the effect those changes had on management success of that program.

Phase I – Historical Data

The goal of this data discovery effort was to acquire a feel for what has been ongoing in the world of software acquisition management over the course of the past 10 - 20 years. The search began with known failed software modernization efforts named in Chapter 1. Searches of the Defense Technical Information Center (DTIC) were conducted to uncover any documentation. This search for historical data can be classified as descriptive research, "a careful mapping out of a situation or set of events in order to describe what is happening behaviorally." (Rosenthal, et.al. 1991)

Inquiries of lesson learned or program data were also made to the Center for Acquisition Excellence, Air Force history offices at the AF, Wing, and Group level, AF Contracting office, system users, and personal contacts. The results of all inquires are discussed in Chapter 4.

Phase II - Survey

A *cross-sectional between-cases design* survey was conducted to gather insight into whether or not program characteristics, such as weapons system vs. support system and complexity level, affect the adoption of the primary recommendations of past defense software studies described in Chapter 2. If the recommendations were implemented what affect did they have on the outcome of the software portion of the program? The cross-sectional between-cases design is the most common form of survey study, where scores on measures of X (independent) and Y (dependent) variables are obtained only once (Schwab, 2005). There are weaknesses to this design, such as a bias introduced with the respondent providing a response for both the independent and dependent variables, and the threat of researcher effects by the way the survey is presented (Schwab, 2005)). Despite these weaknesses, this survey design was chosen due to the time constraints limiting the time available to gather data to only one administration of the survey. A copy of the survey is included in Appendix A.

Sample

The population selected for the survey was Air Force system managers, deputy system managers, system support managers and deputy support managers. Each of these individuals were asked to also forward the survey on to members of their program team responsible for the software portion of their respective program. "The sample frame is a list or set of directions identifying all the sample units in the population." (Alreck, et.al. 2004) The sample frame for this survey was from the Air Force Systems Information Library's "Air Force System Manager Address book". This address book identified 142 system managers and 142 deputy system managers, and 137 system support managers and deputy support managers all of whom were provided the opportunity to respond to the electronic survey.   In addition to these 421 respondents directly targeted, the goal was for these respondents to also forward the survey on to their software management teams as additional potential respondents. Due to time constraints, the electronic version of a survey was deemed the only possible method for administration due to the ability to delivery rapidly and the potential for instant response. Response rate was expected to be equal to a paper based survey if not higher due to the technical aptitude of the respondents. Given their position in a software related field they are anticipated to be technically savvy. Respondents were assured their identities would remain anonymous. They were not asked to identify themselves or their specific program.

Survey Instrument

A 70-question survey was developed to evaluate whether the size and scale of a software or software intensive program affected the implementation of previous defense software study recommendations described in chapter 2.   In addition, the survey investigated whether the implementation of said recommendations affected the program's success or anticipated success in the case of programs still in the early stages of the lifecycle. The survey was comprised of a combination of multiple choice, verbal frequency, forced ranking, semantic differential scale and Likert scale questions. It was designed to take between 10 and 20 minutes. The survey model was developed using Schwab's model (Schwab, 2005.)   This model is a graphical means of representing the relationship between the variables of concern in the research. Figure 3 depicts Schwabs model, while figure 4 is a model of the survey. Conceptual and operational variables are defined as follows: a conceptual variable is a mental definition of an object or event, also referred to as a construct; an operational variable is a variable that is measured to obtain scores from the cases studied (Schwab, 2005.) For example, one conceptual variable in the survey is the program attributes, which may be defined by various characteristics of the program. Individuals may have the same or different mental picture of a program's attributes. The operational variables for this construct as shown in figure 4, are scale, level of complexity, type of system, etc. Each of these operational variables is a measurable indicator of the construct, the program attributes.

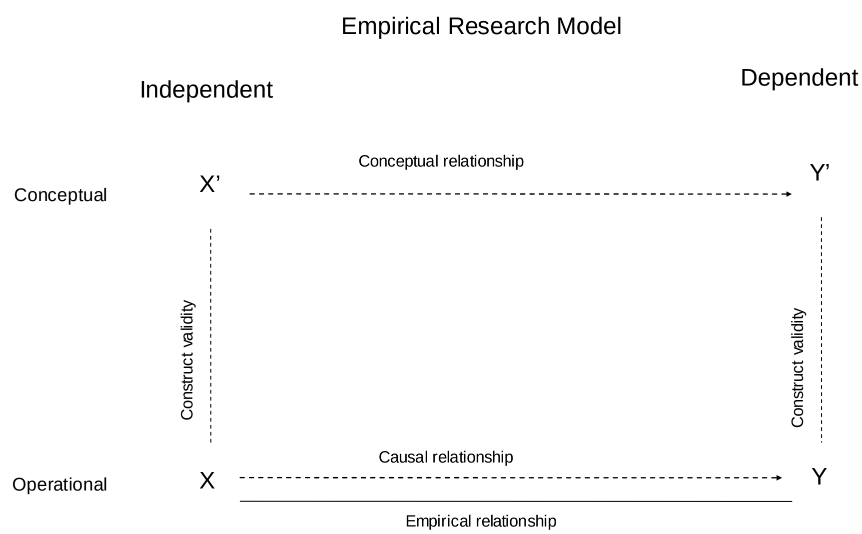


Figure 3 - Schwab's Empirical Research Model

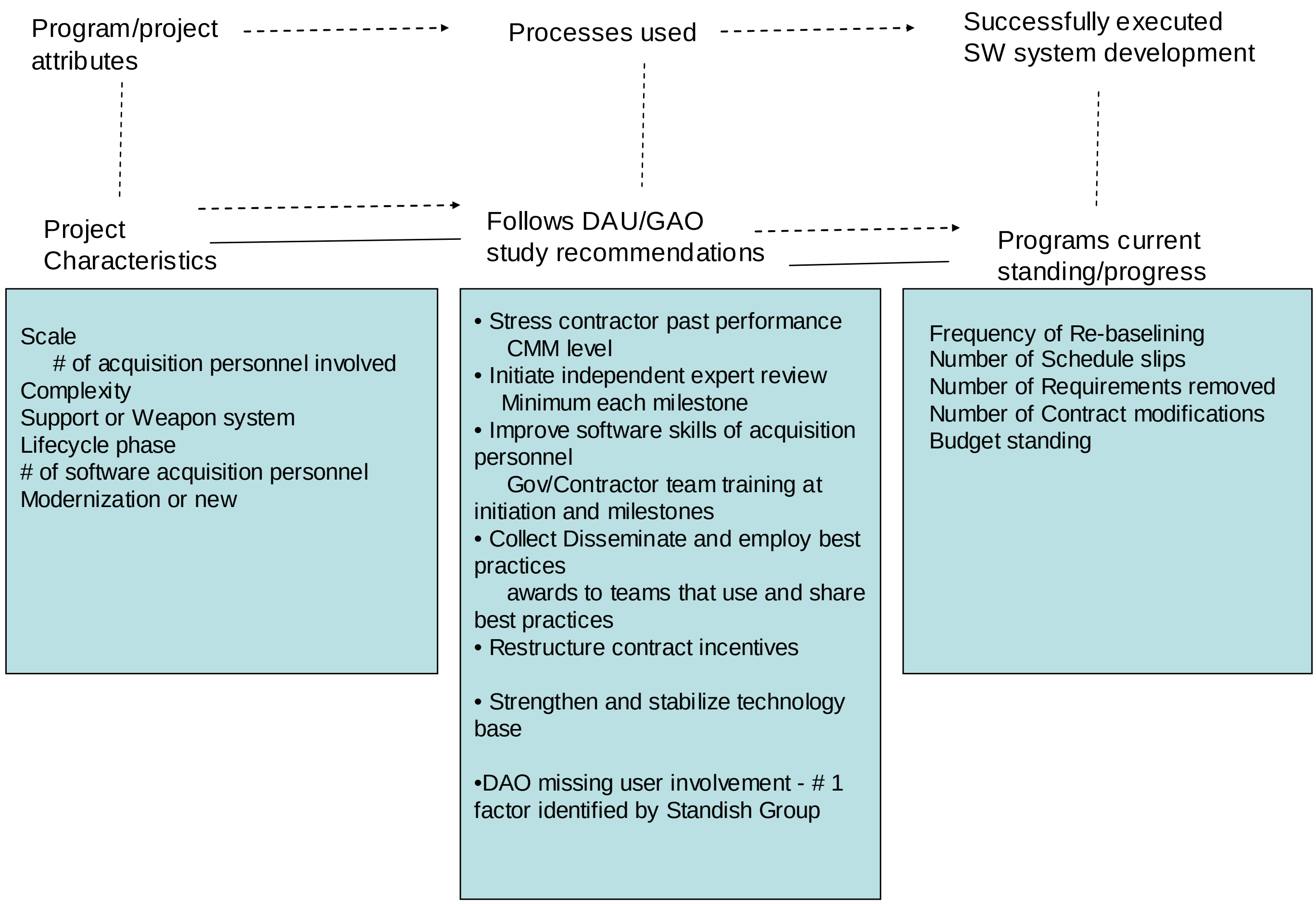


Figure 4 - Survey Model

Constructs Measured

The constructs of the survey as depicted in figure 4 are detailed below. Tables 2 - 4 depict a synthesis of the survey questions to the constructs measured by each.

Program/project attributes

The first portion of the survey contained ten questions to gather data on program attributes. These included questions on scale, complexity, type of system (support or weapon), current lifecycle phase, and whether the program was a modernization effort or a new development. This data was gathered for use in analyzing whether program characteristics influenced the implementation of the DSB recommendations. It is anticipated that the program/project attributes do in fact affect the implementation of past recommendations. We posit the following:

(1) The further along a program is in its life-cycle, the less likely it is that DSB recommendations have been implemented. As a program progresses it becomes more difficult to implement change, especially if it is perceived that the current processes are working well. It has only been 5 years since the DSB report was published and many of these programs have been ongoing for much longer than that, some over 25 years.

(2) Support systems are more likely than weapons systems to have implemented recommendations due to the likelihood that they are of smaller scale and typically have less restricted timelines to complete the project.

(3) Programs that are replacing a large percentage of government software are less likely to have implemented recommended changes. Programs that are largely modernization efforts are focused more on converting old code to new and aren't as compelled to follow guidance aimed at new development.

(4) The complexity, even if only perceived complexity, of a program will increase the likelihood that the DSB recommendations have been implemented. The DSB

recommendations are based on accepted software engineering practices. The managers of more complex systems will be more inclined to accept these engineering practices in order to control a difficult program. The lifecycle phase of the program was gathered with a single selection multiple-choice question, question 15. The amount of time the program has been ongoing was also asked via a fill in the blank. The complexity question gathered the respondents view on the difficulty level of the software being developed. The complexity was collected using an 8-point sliding scale requesting the respondents rating of the complexity of the software where one = simple automation, such as a form program and eight = embedded real time, i.e. fly by wire. Complexity was also measured by the amount of COTS used and was gathered with a verbal frequency question; very little, some, larger percentage or 100%. The type of system was gathered with a multiple choice single answer question with the following choices; weapons, support, or other.

Processes Used

Thirty questions were used to determine if programs had implemented any of the six fundamental recommendations made by the Defense Science Board and detailed in chapter 2. To determine if the programs stressed contractor past performance, the survey included questions about the contractors CMM level and experience. To determine implementation of independent expert reviews (IER's) respondents were asked if IER's had been planned for in their program, if the plan was followed, and how frequently IER's were conducted.

The Defense Science Board identified a lack of software skills among acquisition personnel, to include the program managers, deputy program managers, and all staff involved with the acquisition of software as a major cause of problems; therefore emphasis was placed in collection of data related to this area. The survey included twelve questions to determine the skill level and experience in software systems management and acquisition of the respondents. (See appendix A)

Respondents were asked opinion questions on collection, dissemination, and employment of best practices. These questions focused on the level of effort used to reduce complexity of the software and whether COTS were used to reduce complexity. This area also addressed the collection, analysis and use of metrics in the decision making process of the software acquisition lifecycle.

A final area addressed in this portion of the survey is the amount of user involvement in the software acquisition process. While this was not a finding of the DSB or GAO it is the most influential factor in software success according to a 2001 Standish Group study on software and therefore was included in this research. (Standish Group International, 2001)

Successfully executed software system development

The most difficult section to quantify is the success of the software program. The following attributes were chosen as measures based on data available to respondents and expert opinion of the indicators of problems. Respondents were asked twelve questions to garner a sense of the success or failure of their program to date. These questions addressed the number of times the program/project was rebaselined to date, schedule slips, removal of requirements, contract modifications, software budget standing, changing of requirements, number of engineering change orders, and defect trouble reporting rates. These questions were not specific number questions but rather scale questions. For example question 49 asks “How many times has your program been rebaselined since the initial APB? Never; 1 – 5 times; 5 – 10 times; More than 10 times”.

Table 2 - Synthesis of project characteristic measures to questions

Project Characteristics		
Characteristic	Measure	Question
Scale	ACAT level	14
	Size of acquisition team	22
	Cost	19
	Percentage of overall program	23
Complexity	Real time vs. automation	18
	COTS used	16
Support or Weapons system		24
Lifecycle Phase	Phase	15
	Timing	21

Table 3 - Synthesis of software management success to questions

Program Software management success		
Indicator	**Measure**	**Questions**
Rebaselining frequency		49
Schedule slips		44
	SPI	37
Requirements Removed		45
Contract modifications		40,41
Software budget	CPI	38
Requirements changed		42, 43
Engineering Change orders		46
Defect/trouble reporting rates		48

Table 4 - Synthesis of implementation measures to questions

Implementation of past study recommendations		
Recommendation	Measures	Questions
Stress contractor past performance	CMM level	30
	Experience	25,26,27
Initiate independent expert review	Planned for	34
	Frequency	33
	Actual occurrence	31
Improve software skills of acquisition personnel	Experience	1,2,4,5
	College education	6,8
	Acquisition training	10,11,13
	Software training	7,9,12
Collect disseminate and employ best practices	Reduce complexity	35,36
	COTS used	39
	Use iterative development	47
	Metrics collected	64
	Metrics analyzed	65
	Analysis used in making decisions	68
	Team training	29,31
Restructure contract incentives		28
Strengthen and stabilize the technology base	Government software engineers on staff	54
User involvement	member of acquisition team	20
	user involvement in requirements	19

Statistical Methods

Statistics describe scores on a variable or a relationship between scores on different variables (Schwab, 2005). The responses to the survey were scored as depicted in Appendix C (?). These scores were then analyzed for frequency of scores to individual variables and correlation between the scores for each construct to identify a potential relationship.

To answer the first research question; have the recommendations of the 2005 DSB been implemented? The results of the implementation section were recoded to indicate whether each respondent had or had not complied with the recommendations.

Creation of an Implementation Score for each respondent

For use in later relationship analysis an implementation of past recommendation score was created. This was accomplished by recoding each of the questions outlined above as either a 1, did comply with the DSB recommendation or a 0, did not comply with the DSB recommendation, with the exception of question 36, amount of effort used to reduce complexity in which the scale rating was left in its original format. Individual educational background was not used in the compilation of the implementation Score. The only variables used were the recoded responses to the questions identified in each of the four implementation of past recommendation areas outlined above. The number of recommendations followed as indicated by a score of 1 were summed to provide an implementation score for each respondent. This score was then used as the dependent variable and compared against each of the program characteristic variables to determine if any of the characteristics had an impact on the implementation score the results are described in the Analysis of Relationships section of this chapter. From this point forward this implementation of past recommendations variable will be referred to as the "implementation score".

This is not an uncommon method according to Shore and others 2003, "It is not always essential that the items in summative scales hang together - this depends on your research objectives. If you are trying to say that people who have had more of a certain type of experience are more likely to ... than are people who have had fewer such experiences, then it is perfectly appropriate."(Shore and others, 2003)

In addition to the analysis of whether or not the DSB recommendations had been implemented for each program we wanted to get a feel for the level of software training per individual. Frequency tables were used to view data from the respondents answers to questions about their individual education, acquisition training, computer courses taken and certificates held.

Creation of a Management Success Score for each respondent

The third portion of the survey gathered information about the success of software management in each of the respondents program. The measures chosen for this section are commonly collected metrics used to monitor progress of a software program or project; frequency of rebaseling, schedule changes, SPI, CPI, requirements changes etc. There is little doubt that these indicators have faults, but they do provide a means to measure the effectiveness of methodologies on program success. The measures taken individually are not a good indicator of management success, but combined we feel they can give a feel for the potential success of the program and therefore indicate whether or not the program is being successfully managed.

In order to evaluate the third research question; did implementation of the DSB recommendations have an inpact on the software management success of the program?

The management success indicator variables were recoded to provide a single variable to be used as a management success indicator. This variable, known as the management success score from this point forward, was then be used to evaluate relationship between the program characteristics and management success

Threats to validity

Survey's are always subject to validity threats regardless of the specific design type. (Schwab, 2005) This survey is no exception and we anticipate the following threats will exist:

- Research expectation, based on the description of the research and the wording of questions it is possible a respondent will anticipate the research expectations and answer in a manner they feel is what the research is looking for rather than the factual answer.
- Maturation, of the program and the respondents. Programs that have been on going for a longer amount of time will likely have improved their efficiency over time. Likewise respondents with more experience are likely to be more trained.
- Selection; respondents are self-selected from the population contacted, and therefore may have strong feelings either positive or negative about the subject of software acquisition.

Summary

This research was conducted in two phases. Phase I consisted of a search to for historical data on past software modernization efforts that were failures or experienced extreme delays. Phase II consisted of the development and administration of a survey to determine if recommendations of past software research has been implement and if implementation had an impact on the outcome of the software program/project. Data from both phases was analyzed using the statistical methods described in this chapter. The results can be found in Chapter 4.

IV. Analysis and Results

Results of Search for Failed Modernization Program data

It is common understanding that learning from our mistakes is a valuable tool for becoming more productive in anything we do. This holds true for software acquisition and development. In 2004 the Assistant Secretary of the Air Force (Acquisition) Marvin R. Sambur published a memorandum titled "Revitalizing the Software Aspects of Systems Engineering." In this memorandum he directed that in order to support agile acquisition objectives one of many software focus areas that must be addressed beginning before milestone A and continuing throughout the lifecycle of the program is "Lessons Learned." Each program was directed to support the transfer of lessons learned to future programs by providing feedback to center level Centers for Acquisition Excellence and other affected organizations.(Sambur, 2004) Therefore, this is where the search for data began. The ASC Center for Acquisition Excellence (ACE) was first contacted in March 2005 via phone requesting access to any software postmortem or lessons learned data; to include any type of program data regarding software efforts, with failed software acquisition efforts being of particular interest. All contacts to this organization were answered with similar responses; "this is not something that is tracked by AE." The center of excellence personnel however did suggest that plans and programs (XP) might be a place to find this information. The plans and programs office at ASC was contacted via telephone. Plans and programs personnel stated they did not track historical information on programs or lessons learned. They recommended contacting the Acquisition Center for Excellence.

The search then turned to the ASC/CCX for leads on where to find historical or lesson learned type information on failed software initiatives. The CCX identified ASC/EN as a possible source of this type of data. ASC/EN had recently published the AF software policy memo which directed among other things to “support the transfer of lessons learned to future programs by providing feedback to center level Acquisition Center of Excellence and other affected organizations.” (Sambur, 2004)

A search of DTIC resulted in the location of the original requirements document (ORD) for the ROCC/SOCC modernization. Using this information led to inquires at ACC in an attempt to contact the person or office of the original point of contact on the ORD. The POC office symbol was ACC/DRCW, in the time since this ORD ACC has reorganized and ACC/A8X is the new office symbol for that office. Personnel at A8X also did not have access to any historical data on this system but suggested contacting the Historical Research Agency at Maxwell AFB, AL. A8X did provide insight into the fact that over the years the program name had changed several times from ROCC/SOCC modernization to R/SOCC, to RAOC ADS, to RSGP and finally to BCS-F.

Personnel at the ACC History Office indicated that the “historical records transferred to Tyndall and or Maxwell when 1AF transferred to Tyndall in 1992”, and suggested contacting the 1AF HO for further information (McAlister, 2005).

Contact to both the AF HSO and the 1AF HO resulted in the same answer. These offices simply track histories of operational units and organizations not program data. Both offices suggested DTIC as the source for the information in question.

Individual program offices were contacted via e-mail and phone and the results were similar from all. The most common response was something similar to the following from OSSG/LR "We have lots of lessons learned, but not a whole lot documented."

Since most AF software is produced by contractors the search next moved onto the contracting arm of SAF/AQ. The results were the same, each level inquired stated they did not have any information on the programs in question. Suggestions were always to contact the ACE or search DTIC.

Figure 5 depicts how each major entity queried pointed to another entity as the source of the data. In the end all arrows pointed back to DTIC and SAF ACE. This seems to indicate that the acquisition community is under the assumption that the place to find this type of historical data is in one of these two places. However in reality the data is not stored in any easily accessible single location making analysis extremely difficult if not impossible.

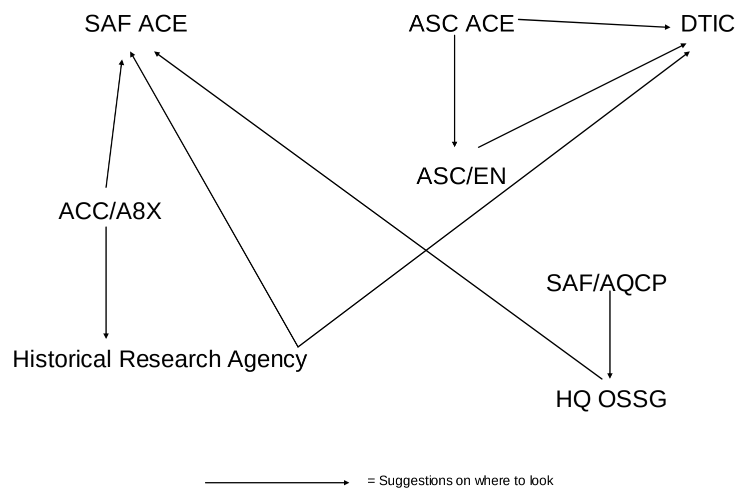


Figure 5 - Graphical representation of data search trail

With no luck at the ASC ACE, inquiries were sent to the SAF ACE via e-mail. One individual familiar with the ROCC/SOCC modernization responded that "There is a lessons learned report from the Litton days, in the files I left in my office" he requested the individuals in his previous office make those lesson learned documents available. This document was never received although extreme effort was not made at this point to acquire the document. One program lesson learned was not going to be enough to analyze trends. The real lesson learned here is that documentation of our failures and set backs, if it even exists, is extremely difficult to find and therefore impossible to learn from.

Results of Survey Analysis

A survey was developed and administered to current AF system managers, assistant managers, system support managers and deputy support managers with a request that each forward the survey on to their respective software teams for team member participation. The survey was sent directly via electronic mail to 421 potential respondents with 18 returned as undeliverable for a total potential respondent count of 403. Of those, only two confirmed that the survey had indeed been forwarded to their respective software teams. Forty-three responses were gathered from this group of potential respondents for a response rate of 10.67%. While a disappointing number it is not unusual for such a low rate to occur in survey studies. According to Alreck and Settle mail surveys usually have a response rate of 5 or 10% and online surveys are subject to the same "very substantial rates of nonresponse and the bias and error associated with it."(Alreck and Settle, 2004:36-37) The low response rate leads to a level of self-selection error in that those who choose to respond are quite often on the extremes of positive or negative feedback on the subject matter.

The purpose of the survey was to investigate if the DSB's recommendations have been implementation. We also looked to determine if system characteristics affected whether a program had implemented the recommendations and if the implementation of past recommendations affects program management success. The DSB indicated that implementation of their recommendations would lead to improved software management success. The goal was to glean information that may indicate in more detail that this was true or indicate other issues affecting the success of software and software intensive programs.

Limitations

The analysis of survey data is limited by the number of responses received. With 43 overall responses and much less in each category only very simple analysis was possible. Some of the relationships would benefit from regression analysis to further define specific relationships, however the small number of responses did not provide enough data for this type of analysis. Despite these limitations, the data did provide insight into issues in the software acquisition community and provides the beginnings of a roadmap to direct future research in this area.

Description of Program Characteristics

The survey sample provided a generally normal distribution of these different program characteristics as can be seen visually in the histograms (figures 6 through 8) below. The only lifecycle phase not represented is the technology development phase however our real area of interest is in programs that are in system development phase and beyond. It is shown in the analysis section of this chapter that the responses from the three programs in concept refinement were eliminated due to missing variable data, these programs were not far enough along for respondents to answer many of the implementation of recommendation and management success factor questions.

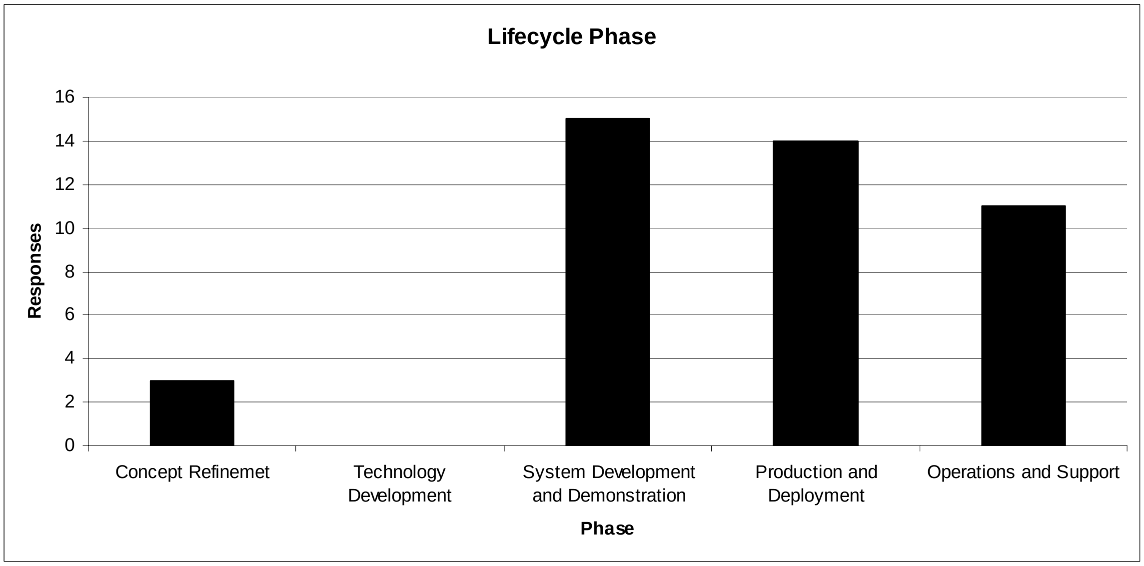


Figure 6 - Lifecycle Phase of responses

Respondents were asked to choose the category their program belonged to, either a weapons or support, the results are shown in figure 7. We anticipate the category of a system will impact the implementation of past recommendations as discussed above.

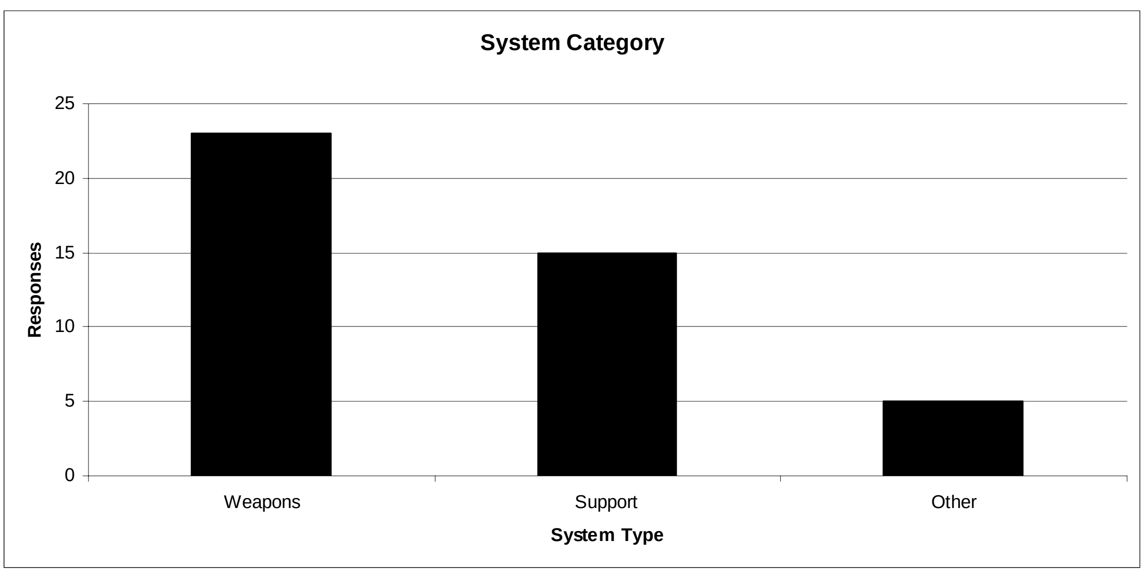


Figure 7 - System category

The primary area of interest of this study was modernization efforts therefore respondents were asked to identify the percentage of their software effort that was replacing an existing government system (or GOTS – government-off-the-shelf). As seen in the figure 8, 70% of the sample reported replacing some existing government software, 26% of the sample reported that 50% or more of their software effort was replacing government software.

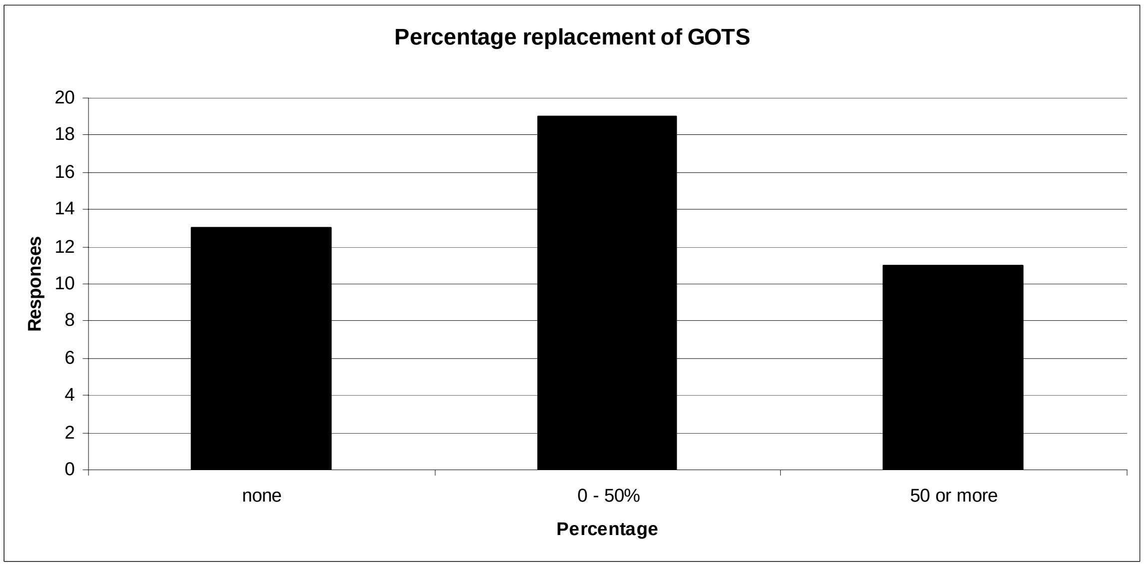


Figure 8 - Percentage replacement of GOTS

Respondents rated the complexity of their system on a sliding scale. The scale ranged from one to eight. One equals the complexity level of a simple automation, for example: a form program, up through eight which equals the complexity of an embedded real time system, for example: fly-by-wire flight controls. The results are depicted in figure 9, 74% of respondents rated the complexity of their system to the right, more complex, portion of the scale as was expected with the sample space being Air Force programs.

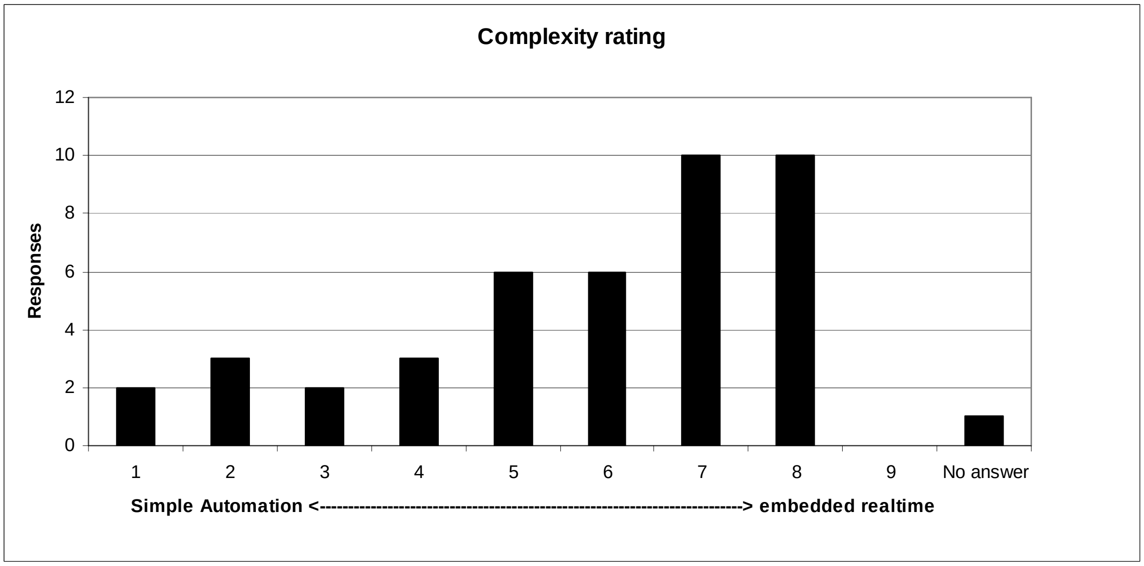


Figure 9 - Program complexity rating

Analysis of the Implementation of Past Study Recommendations

As described on page 14 of chapter 2 the DSB's most recent study published six recommendations to improve the AF's software acquisition process. We asked respondents to answer the questions found in Appendix A, to determine if the following four of the DSB's most recent recommendations have been acted upon. These four were chosen from the six for their ease of measurability. The fifth and sixth recommendations are outside of the scope of the acquisition arena in which this study is focused. The DSB's recommendations for each of the four areas addressed in this research and the survey questions related them are outlined below:

1. **<u>Stress contractor past performance</u>**

“The task force recommends that the DOD strengthen its past performance criteria and restrict program awards to those who have demonstrated successful software development capabilities”(DSB, 2005:ES-2)

DSB recommends:

1. Require all software development contractors to demonstrate CMM level 3 or equivalent processes.
2. Weigh past performance & development process maturity in the source selection process.

Survey questions used to gather related data and responses:

Q30 – CMM level

Q25 – Source selection criteria

Q26 – Contractor completed systems of same scale

Q27 – Contractor completed system in same language

Respondents were asked to indicate the CMM level, an indicator of software process maturity, of their software contractor; the results are depicted in Figure 10. Fifty-six percent of respondents did not know the CMM certification level of their contractor, but of those that did all but one complied with the DSB recommendations.

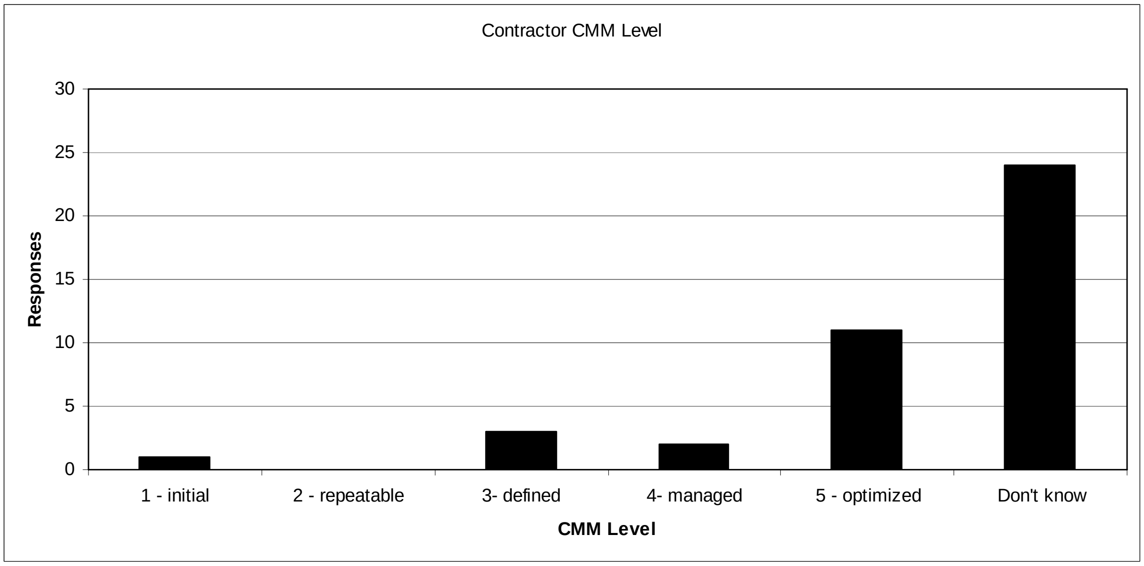


Figure 10 - CMM Level of S/W contractor

Respondents rank ordered, 1 – 6 with 1 being the most important, the importance of contractor attributes used during selection of the contractor for their software program. The results are shown in table 5. Contractor past performance was the one that ranked first overall indicating that the field leans toward the DSB's recommendation of considering the contractors past performance first and foremost during contractor selection.

Table 5 - Ranking of contractor selection criteria (n = 19)

Attribute	Median	Mean	Standard Deviation
Past Performance	2	2.47	1.58
Knowledge of Legacy System	3	3.68	1.60
Proposed Cost	3	3.74	1.45
Proposed Schedule	3	3.16	1.68
Language Expertise	4	3.89	1.45
CMM certification level	4	4.05	1.81

We also asked respondents if their software contractor had completed systems of the same scale and in the same software language. The results are shown in figures 11 and 12 respectively. We felt these measures would give more insight into the contractor's proven experience. A contractors past performance is a more valid indicator of future performance if those past programs were of similar scale and in the software language of the system being considered for contract.

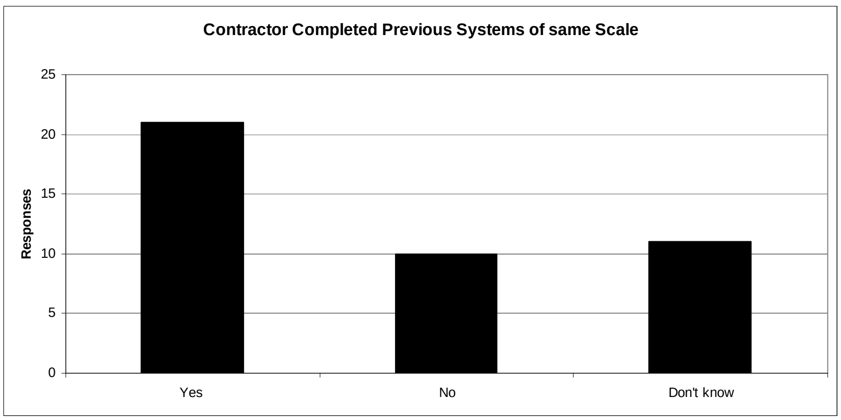


Figure 11 - Contractor completed systems of same scale

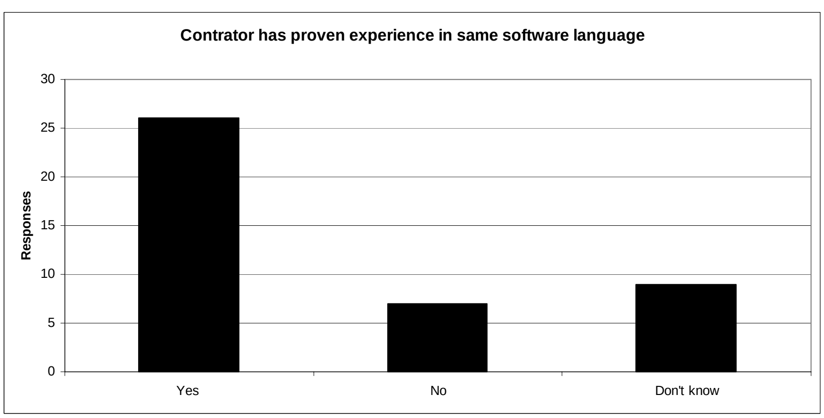


Figure 12 - Contractor has proven experience in same language

The results of these four questions indicate that contractor past performance is considered by the respondents to be an important criteria in contractor selection. Of those that did know their software contractors CMM level all but 1 were rated level 3 or higher, this follows the DSB recommendation. The respondent answers indicate that past performance is the overall highest ranked criteria for contractor selection and that the majority of programs of those that responded have contractors with proven experience.

2. <u>Initiate Independent Expert Review</u>

"Recommend institutionalizing [IER's] on DOD ACAT I – III software-intensive programs"(DSB,2005:20)

DSB Recommends:

1. IER's should be held at key program milestones
2. IER's should be held at least every six months

Survey questions used to gather related data and results :

Q32 **–** Has the program undergone IER

Q33 **–** How often are IER's conducted

Q34 **–** Is this consistent with IER schedule

The DSB recommended conducting IER's for all ACAT levels and recommended they be conducted at key program milestones or every six months. (DSB,2005:ES-3) The survey gathered data on IER occurrence, frequency and how that frequency matched up to the plan for IER's. The results were as follows:

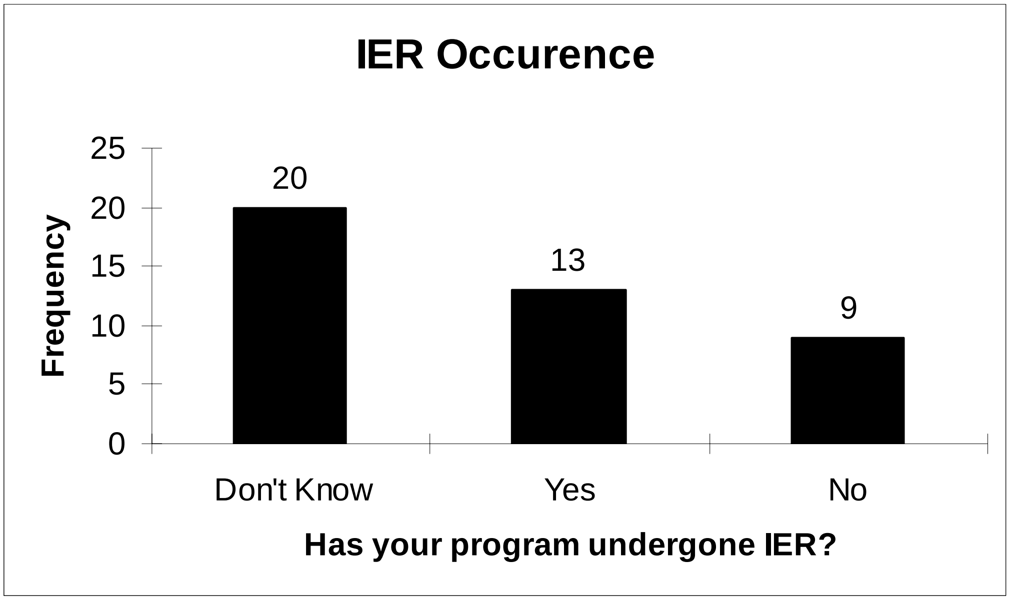


Figure 13 - IER Occurrence

Of the 42 respondents who answered these questions only 13 responded that their program had in fact undergone and IER, however with the high rate of "don't know" answers it is impossible to say whether most programs are conducting IER's. Of those that reported IER's had occurred the majority were conducted within the DSB recommended guideline for frequency, at each milestone or every six months, as seen in figure 14 all but 3 are within this range. The responses for the two "other" answers were, "at each rocket launch" and "occasionally."

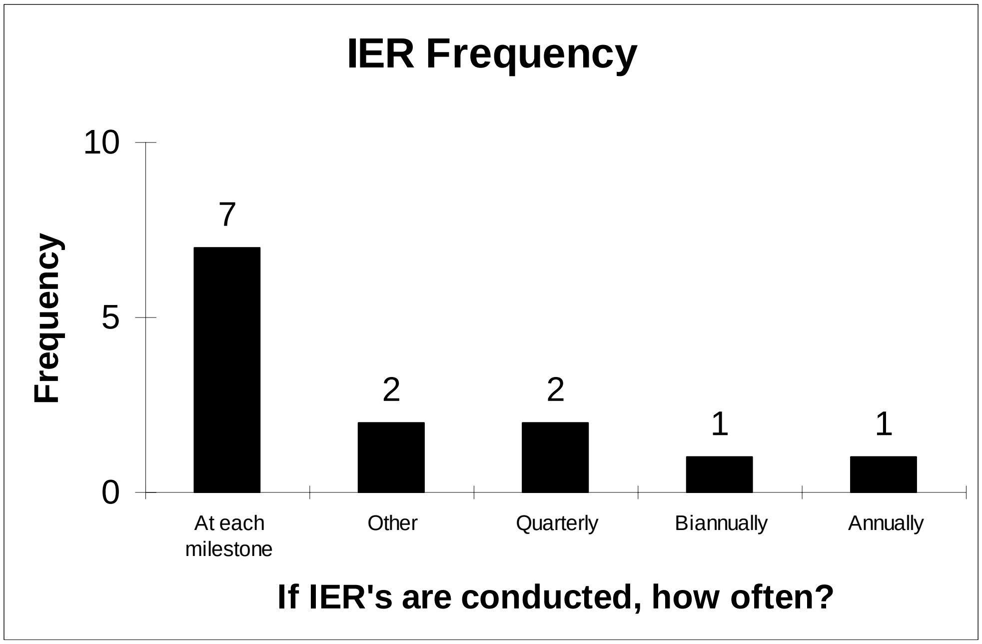


Figure 14 - IER frequency

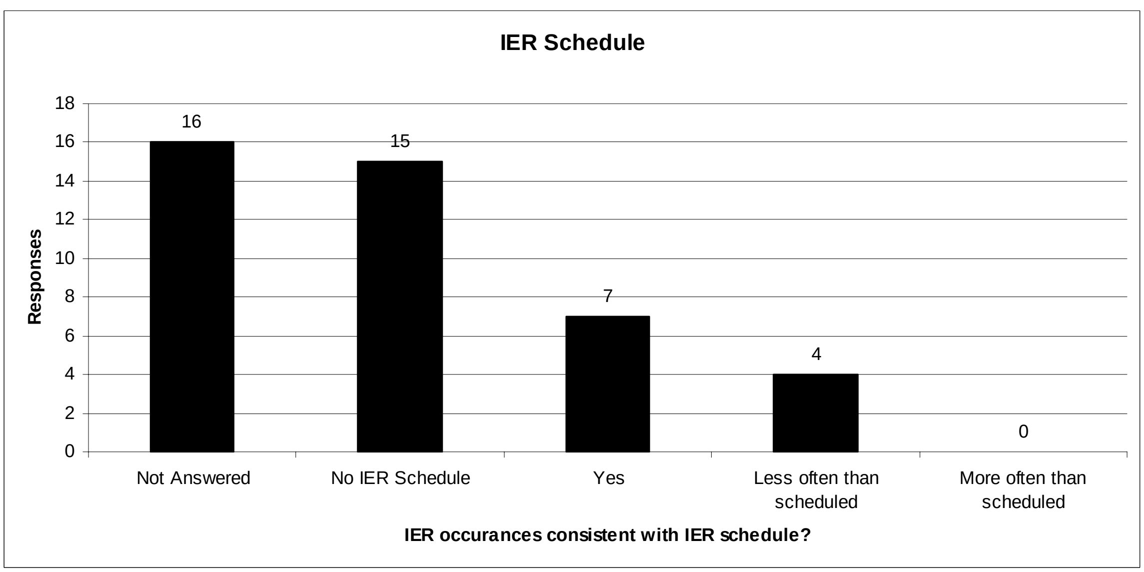


Figure 15 - IER schedule

3. **Collect Disseminate and Employ Best practices**

"The task force strongly endorses the following best practices;"(DSB, 2005:ES-3)

DSB Recommends:

1. Iterative design/development
2. Reduce complexity

Survey questions used to gather related data:

Q47 – iterative development

Q36 – Effort to reduce complexity

The DSB recommended that the DOD collect, disseminate, and employ best practices. Their recommendations included some best practices they "strongly endorsed."(DSB, 2005; ES-3) The survey asked questions related to two of these endorsed practices; iterative development and requirements trade-off. To determine whether or not the recommendation to collect, disseminate and employ best practices has been incorporated into acquisition programs we asked respondents the amount of effort used to reduce complexity, the amount of COTS used, was team training accomplished, and does the program follow iterative development processes.

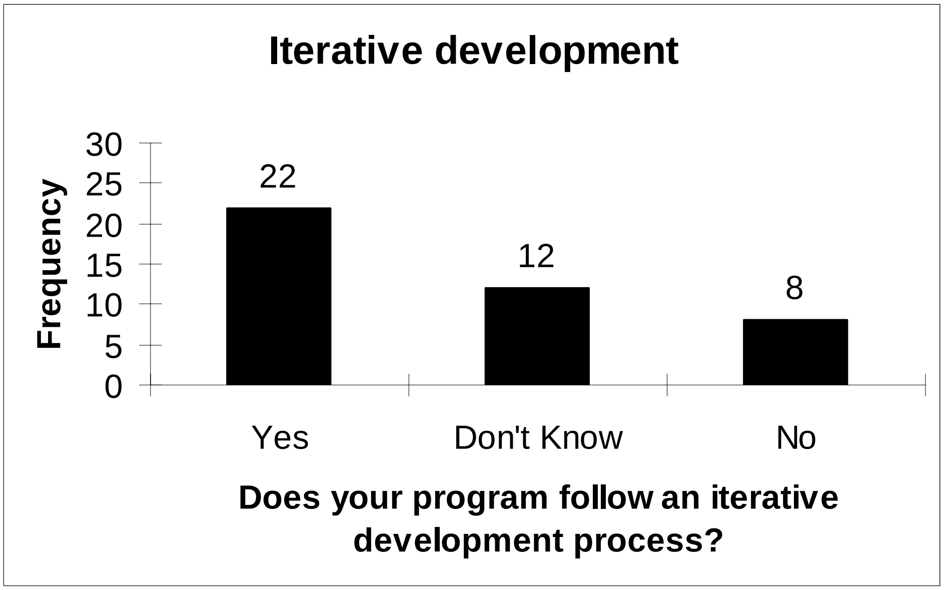


Figure 16 - Iterative development

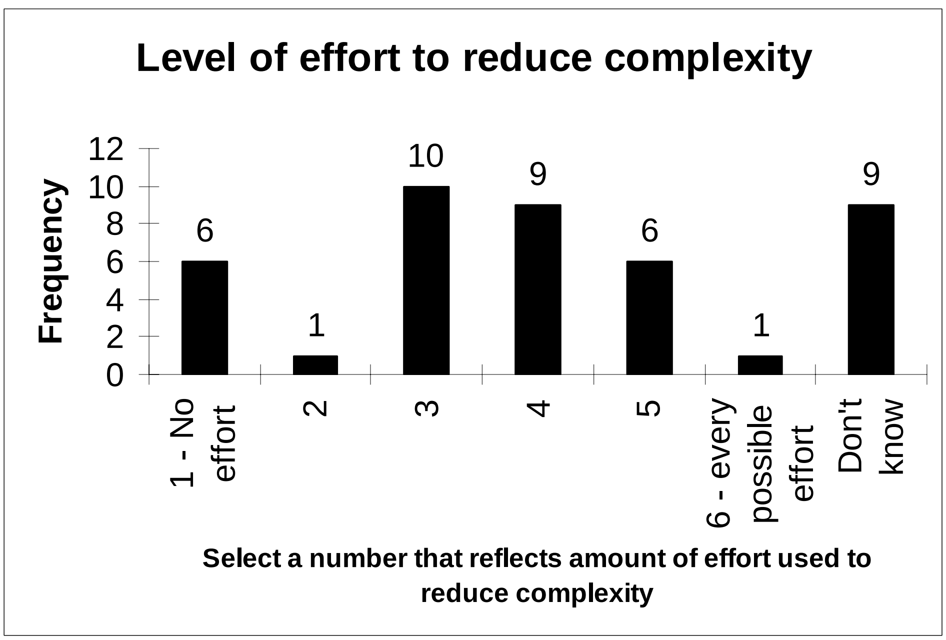


Figure 17 - Reduce complexity

The DSB did not make recommendations on ways to collect and disseminate best practices only that it should be done.   They did however mention better use of metrics and identified core metrics to be collected. These core metrics are the measures we used to give each respondent a management success score in the following section.

4. Improve Software Skills of Acquisition Personnel

"Inexperience and/or unqualified personnel at all levels are a major contributor to DOD software problems"

DSB Recommends:

1. Institute mandatory software intensive systems training for program managers and key staff on all ACAT programs.

2. Require collaborative gov't contractor team training at program start and milestones

Survey questions used to gather related data:

Q12 – Software related training courses

Q11- Software Acquisition training

Q29 – Team training at program initiation

Q31 – Team training at program milestones

The DSB recommended that the DoD improve the software skills of acquisition and program management. They suggested requiring mandatory training of program managers and key program staff before program initiation along with mandatory gov't contractor team training at program initiation and key milestones.

The survey addressed whether respondents pursued further education in software through the Defense Acquisition University offerings or other means. The results are shown in figures 18 through 20. It is surprising that the most attended DAU class, ACQ 201 parts A&B, was attended by only half the respondents. This course is geared toward mid-level acquisition professionals to prepare them to work on integrated product teams by teaching system acquisition principles and processes. It does not specifically address software (DAU Course descriptions). The courses of greatest interest to this research are the software acquisition management (SAM) and information systems acquisition management (IRM) courses which according to the course descriptions cover;

> SAM : "*Covers software acquisition/development risks, DoD regulatory and technical frameworks, software and system architectures, and software development life cycle and integration processes. Software standards, measurements, testing, security, quality issues, process maturity, as well as "best practices" for the management of software-intensive systems are also reviewed."*

> *IRM: "DoD information systems acquisition management. It covers software acquisition/development risks, DoD regulatory and technical frameworks, software and system architectures, and software development life cycle and integration processes. Software standards, measurements, testing, security, quality issues, process maturity, as well as best practices for the management of software-intensive systems are also reviewed."*

As shown in figure 18 only nine of the 43 respondents completed IRM 101 and only three completed IRM 102. Ten of the respondents completed software acquisition management SAM 101 and 5 completed the SAM 201.

The survey also asked if any other software related courses were completed by the respondents. Nearly half, 20 of 43, stated they had completed 'other' software courses. Of those 20 , the majority completed 3 or less other software courses as seen in figure 19.

The recommendation of the DSB was that all government/contractor teams complete annual software technology refresh training, team training at start and at critical milestones, and mandatory software-intensive systems training. These numbers indicate that this recommendation has not been implemented across the board.

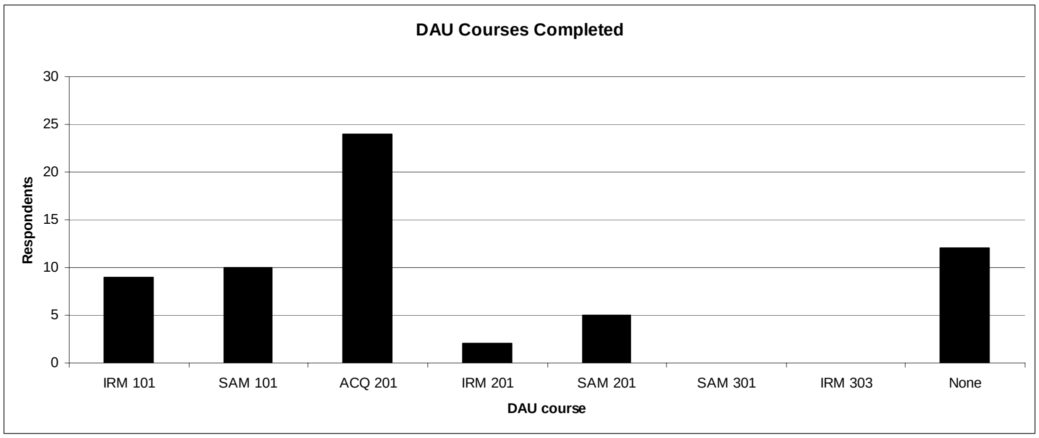


Figure 18 - DAU courses completed

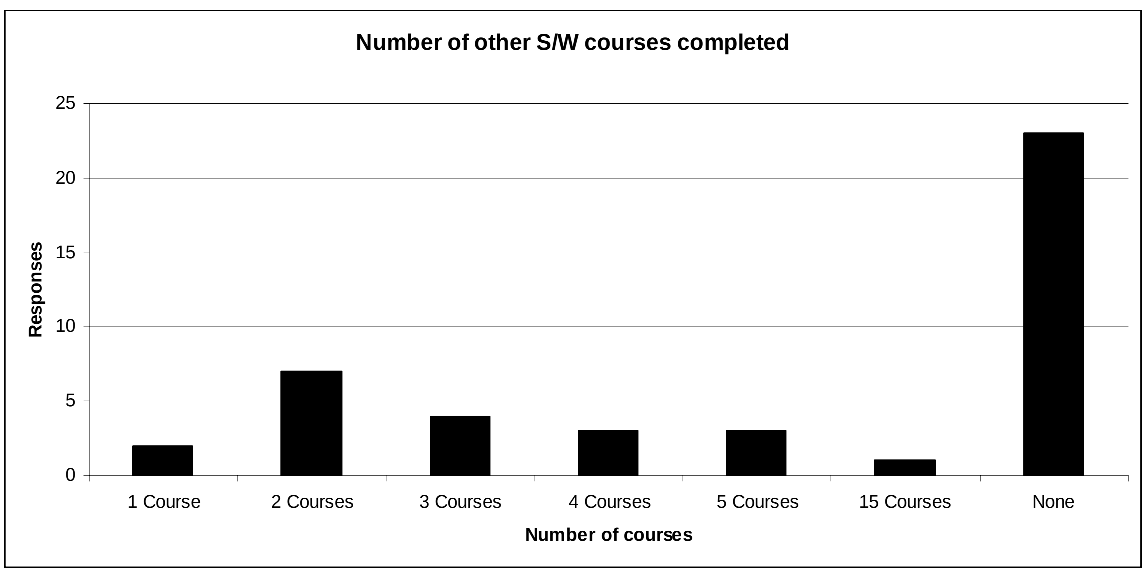


Figure 19 - Number of other software courses completed

Respondents were also questioned about certifications held. The results are shown in figure 20. Only 4 of the 44 respondents held an Information Technology Level I certificate. Given that each respondent is a member of software or software intensive acquisition program, this is another indicator that the DSB and GAO studies findings are still true today. The acquisition community remains insufficiently educated in the field of software management.

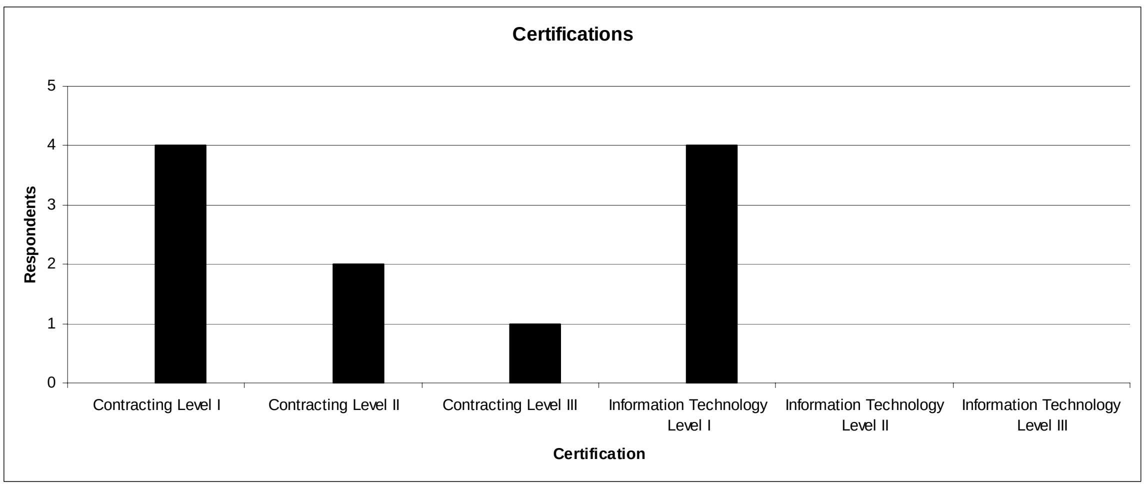


Figure 20 - Certifications held

As seen in Tables 10 through 17 of Appendix B many of the questions gathered information of the level of each respondent's software and acquisition skills. A primary finding reported by the DSB and other past reports as identified in Chapter 2 is that acquisition personnel are not properly trained in the acquisition of software. The hope was to provide evidence to support or dispute these reports. A quick look at the frequency tables of these questions gives the impression that these past findings are still true today.

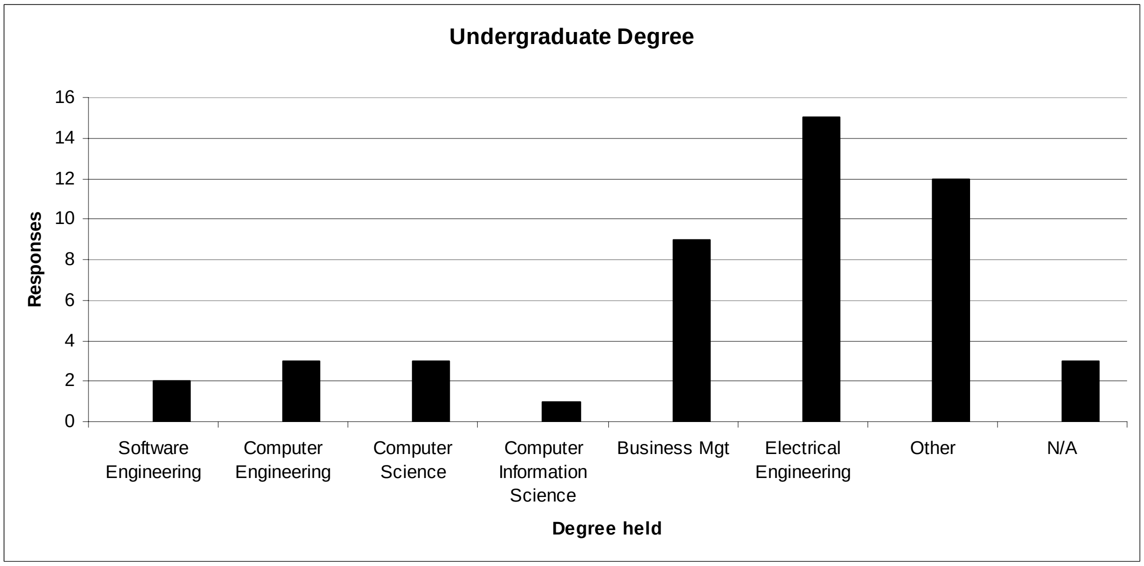


Figure 21 - Undergraduate degree held

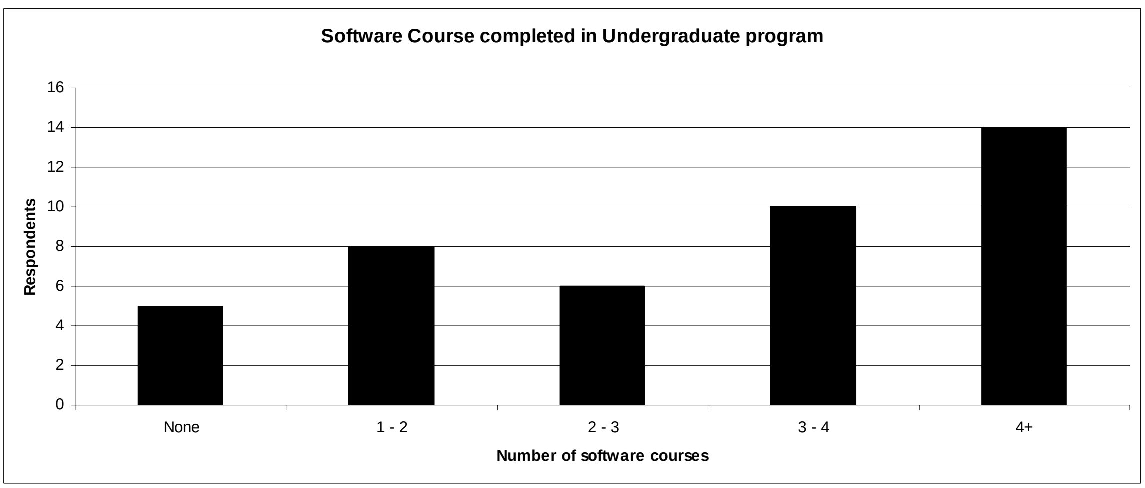


Figure 22 - Graduate degree held

Figure 21 shows that of the 43 respondents only nine indicated they hold undergraduate degrees in a software related major (computer engineering, computer science, etc.) the numbers decrease further when you look at graduate degrees, figure 22, where only four out of the 43 have a software related degree. The survey also queried the number of software courses completed in pursuit of undergraduate and graduate degrees the results are shown in figures 23 and 24. These numbers are more encouraging and indicate that the majority have had at least minimal exposure to software education.

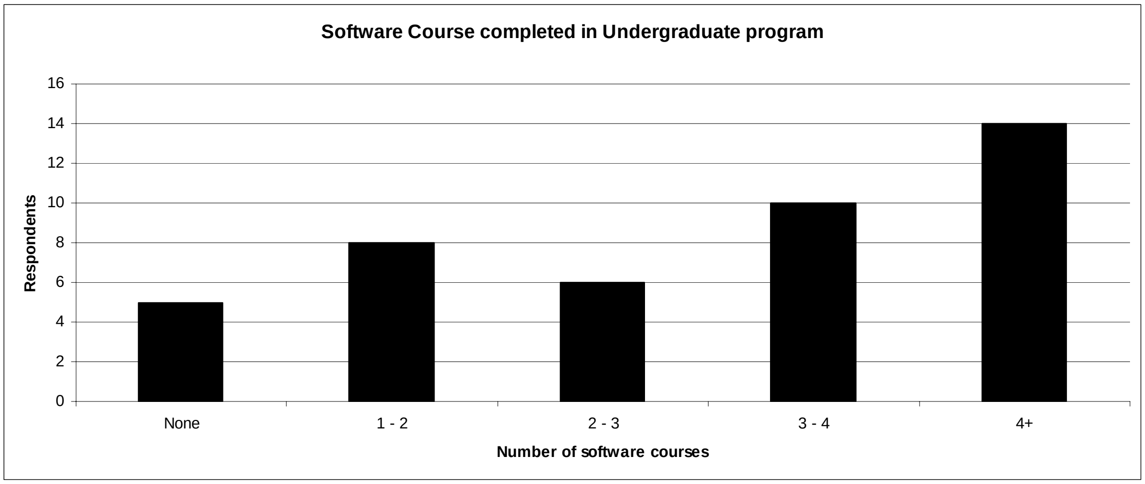


Figure 23 - Software courses completed in undergraduate program

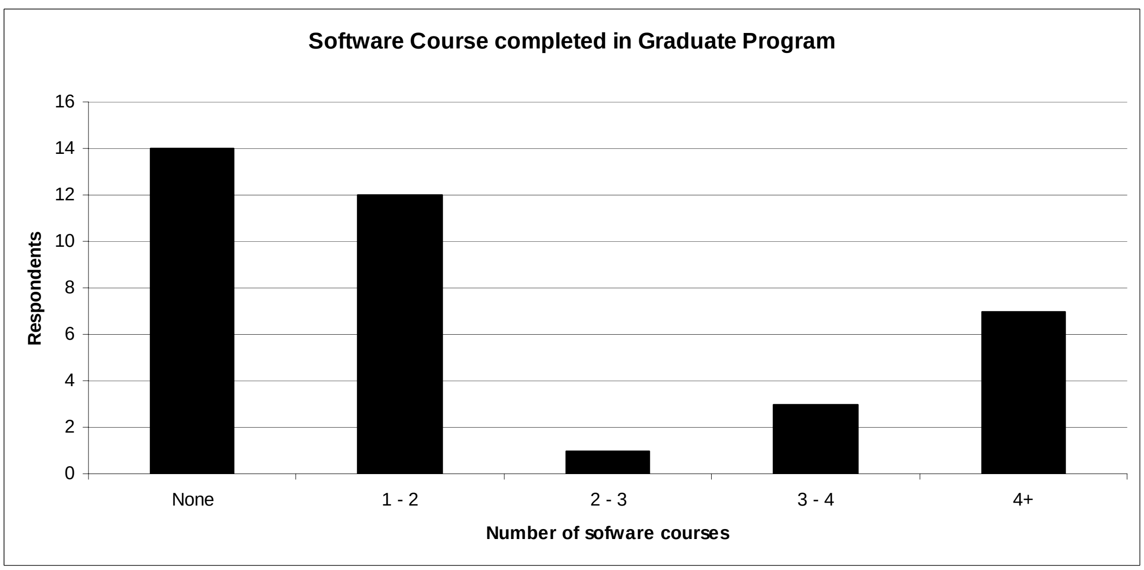


Figure 24 - Software courses completed during Masters program

Analysis of Program Software Management Success

Pearsons correlation of all management success variables showed a relationship between all of them other than CPI. Therefore CPI was removed from the analysis. The management success variables were then recoded as described in tables 18 - 20 of Appendix B. This had the affect of giving a higher value to those programs that indicated more success in each area. The new recoded values were summed to provide a single management success score for each case. This score was used to analyze the relationship between management success and the program characteristics, and between management success and the implementation of past recommendations. The following questions were used to generate the management success score for each respondent:

Q49 Frequency of rebaseling

Q44 Schedule changes due to software

Q37 SPI

SPI is a cost measure related to earned value. SPI is calculated by dividing the planned cost of work performed (or EV: the earned value) by the planned cost of the work scheduled (PV.) (Mantel and others, 2005: 239)

Q45 Requirements removed to adjust for software

Q40 Number of software contract changes

Q41 Number of contract modifications after initial testing

Q42 Requirements changes since APB

Q43 Software requirements changes after initial testing

Q46 Number of engineering change orders executed

Cases that had 4 or more “none” responses were eliminated from the analysis. This had the effect of removing those programs that were not far enough along in their lifecycle to have gone through initial testing. The three programs in the concept refinement phase were eliminated along with others that either were not far enough along to answer these questions or the respondent simply did not know the answers to enough of them.

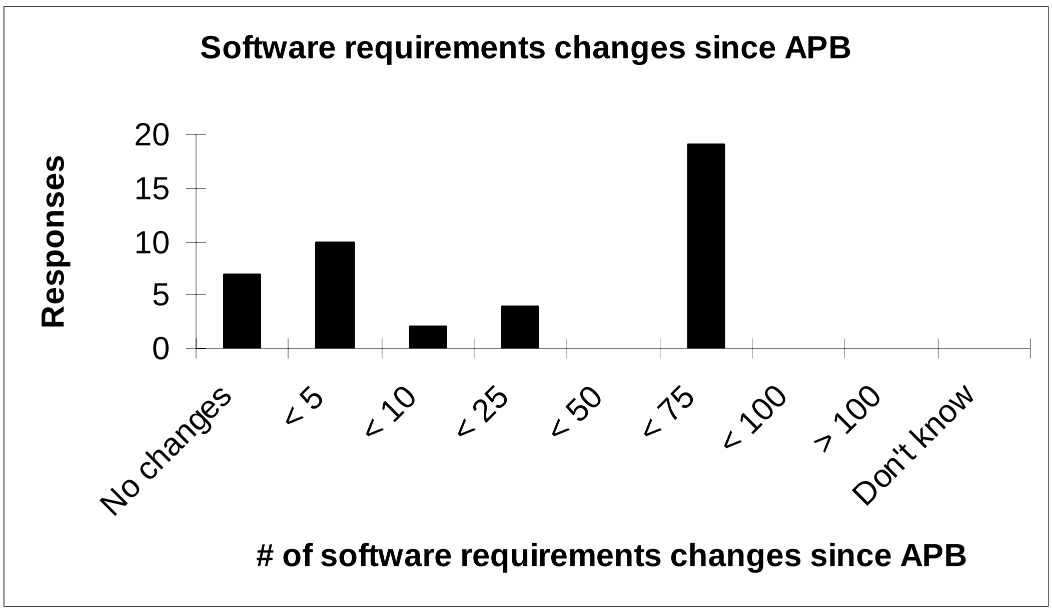


Figure 25 - Software requirements changes since APB

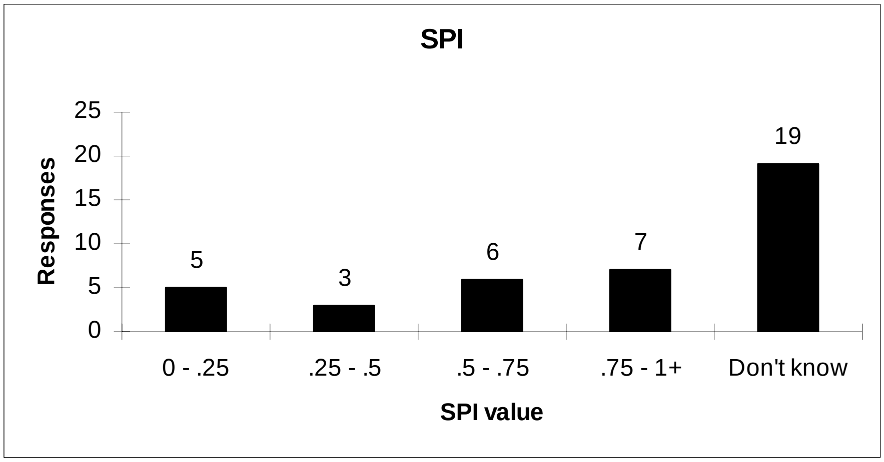


Figure 26 – SPI

Analysis of Relationships

To evaluate if program characteristics affect the implementation of past recommendations and management performance, the program data was analyzed for correlations between the characteristics and the implementation and management scores. The only program characteristic significantly correlated with the implementation score is the system type, weapons or support (table 6). The correlation is as expected; the correlation shows that weapons systems are less likely to have implemented the recommendations than support systems. Weapons systems were giving the higher value; 1 = weapons, 0 = support.

Next we looked at to see if there was correlation between the implementation score and the management success score. We also looked for correlation between the implementation score and the management score. As shown in table 6 the only correlation between the management score and any of the variables or implementation score is a -.503 correlation with the size of the team. This indicates that as the size of the team increases the probability of management success decreases. This result is limited by the small n, but is a candidate for future research.

Table 6 - Correlation table - Management Success Score with Implementation Score and Program Characteristic variables

	Implementation Score	Mgt Success Score	Mean	Std Dev
Implementation Score	1	-0.284	9.32	3.18
Mgt Success Score	-0.284	1	18.3	6.58
ACAT Level	0.058	-0.338	1.76	0.97
Size of Team	0.102	-.503(*)	1.44	0.75
Percentage of cost for S/W	-0.053	-0.154	2.33	0.92
S/W intensive	0.243	0.068	0.69	0.47
Complexity scale rating	0.372	0.333	6.48	1.55
Support or Weapons system	-.556(**)	0.392	1.29	0.46
Lifecycle phase	-0.050	-0.109	3.48	1.01
Duration of program	0.011	-0.008	8.80	11.99
How much is replacing GOTS	0.115	0.044	2.00	0.73

* $p < .05$; ** $p < .01$ (2-tailed).

Investigative Questions Answered

This thesis looked at three investigative questions:

1. Has the Air Force implemented any of the past recommendations made by DOD agency task forces on software? Of the twelve variables used to acquire a feel for the amount of the DSB recommendations that are implemented in the field only, half of them had implementation rates above 50% in the survey sample, as seen in table 7. Acquisition teams seem to be stressing contractor past performance as shown by the high percentages of implementation in all three of the measures for ‘stress contractor past performance’ questions 25, 26 and 27. However independent expert reviews and team training do not share the same high level of implementation as indicated by the results of

questions 32, 33, and 34. The level of training is also not meeting the DSB recommendations as evidenced by the results of questions 29 and 31. Individual training also seems to be lacking some as seen in question 12.

In addition to the measures used to determine the level of implementation of DSB recommendations we also looked at the individual software skills of acquisition personnel. Again we see a similar trend to that reported in the 2005 DSB software report. The field has very limited numbers of software-trained personnel and those without a formal software educational background also appear to not be taking advantage of software training available to them.

Table 7 - Implementation of past recommendation results

Question	Percent of all respondents	Percent of respondents who answered	n
12 - Completed Software related training courses	46.50%	46.50%	43
25 - Contractor past performance most important attribute for selection	27.91%	100.00%	12
26 - Contractor has experience of same scale	48.84%	67.75%	31
27 - Contractor has experience in same language	60.46%	83.87%	33
29 - Gov't/contractor team training conducted at program initiation	11.63%	31.25%	16
30 - Contractor CMM level 3 or higher	39.50%	94.44%	18
31 - Team training at milestones	2.33%	2.33%	42
32 - IER's conducted	30.23%	30.23%	42
33 - IERs Concducted at milestones or every 6 mos.	23.26%	55.55%	17
34 - IER's consistent with schedule	16.28%	26.92%	26
47 - Iterative development	51.16%	73.33%	30
36 - Effort to reduce complexity (mean)	3		34

2. Do the characteristics of a system effect whether software acquisition management recommendations for improvement are implemented? The survey results indicated that the type of program, weapon or support, does affect the implementation of previous recommendations. Weapons systems were shown to be more likely to implement the recommendations.

3. If the recommendations to improve software acquisition management made by the DSB were implemented, did they positively affect the outcome of the program? The results of the survey were inconclusive for this question. The results imply that implementation of the recommendations did not have an effect on the program management success, however much more research is need to make this claim. Management success is something that needs a longitudinal look. A program that appears to be successfully managed by today's metrics may appear completely the opposite in a week or a month. The only way to measure management success somewhat effectively is to look at the trend over a period of time. In addition to a longitudinal look research containing a larger sample is need to make any conclusive judgments about the positive or negative affects of these implemented changes.

Summary

Attempting to locate program data on troubled Air Force software modernization efforts made apparent a knowledge management problem in the acquisition community. Each office or agency pointed to another as the place to find the information, with all arrows eventually pointing back to either DTIC or the ACE. Data that is required to be filed with DTIC is not always filed with them, and DTIC has no way to enforce the policies that are in place that do dictate the data be provided to them. The ACE personnel contacted in the course of this research did not feel it was the ACE's role to track this type of data.

The survey indicated that the recommendations of the 2005 DSB report on software have, at most, only been partially implemented. The area that appears to be most widely implemented is stressing contractor past performance, where the majority of respondents that answered stated they considered contractor past performance to be of high importance for selection. The area that appeared to be in the same state as reported by the DSB is the area of software education of the acquisition community. The survey was inconclusive as to whether or not the implementation of recommendations had an impact on program management success.

V. Conclusions and Recommendations

Conclusions of Research

The first and most significant finding of this research is that the Air Force software acquisition community does not have a process in place for widespread sharing of best practices or lessons learned. This research began with the intention of comparing known failed software modernization efforts in an attempt to uncover similarities that would indicate potential areas for improvement to enhance the ability to more effectively acquire and manage software, especially for modernization efforts. However, this proved to be impossible due to a lack of proper historical documentation on all programs. We believe the data exists but it is not managed in a way to make it accessible when needed.

Next the research identified that the recommendations of the DSB, and subsequently the studies prior to it, appear to have largely not been implemented with the exception of stressing contractor past performance during source selection. The other recommendations reviewed in this research via the survey; initiate independent expert reviews, collect, disseminate and employ best practices, and improve software skills of personnel, all showed low levels of implementation when measured by implementation of the DSB's recommendations for each.

Finally, the research was inconclusive in determining whether implementation of the DSB's recommendations had an effect on the management success of the software program. Since many factors affect the success or failure of an acquisition program, a more in-depth study of this question would need to be done. The data collected in this study however, points toward a possible indication of association between implementing past recommendations and the success of the management .of the program, but the causal direction of this relationship cannot be determined with the available data.

Recommendations for Action

This research showed that the Air Force acquisition community lacks an efficient methodology for documenting and sharing lessons learned. There is plenty of guidance on what needs to be done, but no enforcement of the policies. There is a need for a knowledge management plan to document software program lessons learned. A key to becoming more effective and efficient at acquiring software is to learn from past experiences, but without a knowledge management system in place it is difficult if not impossible to learn from the lessons of past programs.

The software acquisition community would benefit from the development of a knowledge management plan for recording past program lessons learned to enable the sharing of experiences, both good and bad. The community would also benefit from further investigation into what makes a successful Air Force software program. There has been much research into the makings of successful civilian software programs and development of software practices, however the military environment is different and warrants its own separate look.

There is a need for recruitment of software knowledgeable personnel to the acquisition career fields. As seen in the results of the survey, the software acquisition community appears to have minimal personnel with software expertise. While there are software training ???

Recommendations for Future Research

The software acquisition community would benefit from further research into what the inhibitors are within the DoD to adopting recommended changes. Further research into why the acquisition community does not take advantage of the training that is available to them through the DAU could include questions like: Is the training not viewed as worthwhile? Do the courses need updating?

Summary

This research merely scratched the surface of discovering what factors cause Air Force software programs to become dinosaurs that live beyond their originally intended system lifecycle due to an inability to modernize them. A key to finding more specifics is historical software program data, which today is extremely difficult if not impossible to obtain. The obstacles encountered in the search for historical data for this study highlighted a knowledge management problem in the Air Force Acquisition community that deserves attention. It is crucial for success to learn from past failures, as well as successes.

The survey provided insight into what is really occurring in the field. It generated supporting data for several of the major findings of the 2000 DSB report.   The data indicated that we do in fact have a lack of sufficiently trained software experts working on software acquisition initiatives. Additionally, for unknown reasons, programs have not adopted independent expert reviews of their programs, a practice that has led to much success in commercial software development.

Appendix A - Survey

Purpose: This survey is part of a research effort focused on identification of the Air Forces roadblocks to software modernization. This data is necessary for insight into why the Air Force struggles to modernize many of its legacy systems. Our goal is to understand how individuals in the program management team view their role in software acquisition and their opinions on the Air Forces method of managing software acquisition. This survey will help us gauge the implementation of software acquisition practices and understand employees' views on formal and informal software management processes.

Confidentiality: We greatly appreciate your participation in this survey. Your perceptions and actual experiences in working as part of an acquisition team on a program that involves software are essential. ALL ANSWERS ARE STRICTLY CONFIDENTIAL and, unless you wish to tell us your identity, all answers are anonymous. No one outside the research team will ever see your questionnaire. No identification of individual responses will occur. Findings will be reported at the group level only.

Disposition: Results will be published as part of an AFIT Masters thesis, and will be available to you upon request.

Time Required: It will most likely take about 10-15 minutes to complete this survey.

Contact Information: If you have questions or comments about the survey contact Capt McClamma or Lt Col Halloran via any of the contact methods provided below. Thank you very much for your participation.

Sincerely,

DYAN E. MCCLAMMA, Capt, USAF
Masters Student, Air Force Institute of Technology
Air Force Institute of Technology
2950 Hobson Way
Wright-Patterson AFB, OH 45433-7765
DSN (937) 785-3636 X
Comm (937) 255-3636 X
e-mail: dyan.mcclamma@afit.edu

MICHAEL T. REHG, PhD
Assistant Prof of Management
Department of Systems and Engineering Management (ENV)
Air Force Institute of Technology
2950 Hobson Way
Wright-Patterson AFB, OH 45433-7765
DSN 785-3636 ext 4574
Comm (937) 255-3636 ext 4574
e-mail: michael.rehg@afit.edu

PRIVACY NOTICE

In accordance with AFI 37-132, Paragraph 3.2, the following information is provided as required by the Privacy Act of 1974:

Authority: 10 U.S.C. 8012, Secretary of the Air Force; powers and duties; delegation by; implemented by AFI 36-2601, Air Force Personnel Survey Program.

Purpose: To obtain information regarding the management of software in all acquisition programs.

Routine Use: A final report will be used to analyze the Air Forces implementation of the recommendations of past Defense Science Board and General Accountability Office studies of ways to improve the management of software in acquisition programs. No analysis of individual responses will be conducted and only members of the research team will be permitted access to the raw data. Reports summarizing trends in large groups of people may be published.

Participation: Participation is VOLUNTARY. No adverse action will be taken against any member who does not participate in this survey or who does not complete any part of the survey.

General Information

1. How many years (months if less than 1 year) experience do you have working in an acquisition position/function?
___________ years _________ months

2. Approximately how long have you been a member of your current program acquisition team?
___________ years _________ months

3. Which describes you?
   - ☐ Active duty Military
   - ☐ Government Civilian
   - ☐ Other (specify)_______________________

4. How many different programs have you been a member of the acquisition team for (counting your current program)?
____

5. For each item below, indicate on the corresponding line a number from the scale that indicates your level of expertise in that area :

Scale

Novice **1 2 3 4 5 6** Expert

Software ______

Hardware ______

Acquisition ______

Contracting ______

Program management________

6. Check all for which you hold an undergraduate degree:

- ☐ N/A
- ☐ Software engineering
- ☐ Computer Science
- ☐ Information Resource Management
- ☐ Computer Information Science
- ☐ Business Management
- ☐ Electrical Engineering
- ☐ Other (specify)______________________
____

7. How many software related courses did you complete in pursuit of an undergraduate degree (circle)?

None   1 – 2   2 – 3   3 – 4   4+

8. Check all for which you hold a graduate degree:

- ☐ N/A
- ☐ Software engineering
- ☐ Computer Science
- ☐ Information Resource Management
- ☐ Computer Information Science
- ☐ Business Management
- ☐ Electrical Engineering
- ☐ Other (specify)______________________
____

9. How many software related courses did you complete in pursuit of a graduate degree (circle)?

None   1 – 2   2 – 3   3 – 4   4+

10. Have you completed any software specific acquisition courses in the last 5 years?

- ☐ Yes
- ☐ No

If yes, how many? ________________________

11. Check by any of the following defense acquisition university (DAU) courses you have completed:

- ☐ IRM 101 Basic Information Systems Acquisition

☐ SAM 101 Basic Software Acquisition Management
☐ ACQ 201 (Parts A & B) Intermediate Systems Acquisition
☐ IRM 201 Intermediate Information Systems Acquisition
☐ SAM 201 Intermediate Software Acquisition Management
☐ SAM 301 Advanced Software Acquisition Management
☐ IRM 303 Advanced Information Systems Acquisition

12. Have you completed any other software related course? (non-degree seeking, certificate, certification, etc)?
☐ Yes
☐ No
If yes, how many? ____

13. Check by any of the following certifications you currently have?
☐ Contracting Level I
☐ Contracting Level II
☐ Contracting Level III
☐ Information Technology Level I
☐ Information Technology Level II
☐ Information Technology Level III

Program Information

Answer the following questions considering only your current program.

14. The program you are currently working is an:
☐ ACAT I
☐ ACAT IA
☐ ACAT IAM
☐ ACAT IAC
☐ ACAT II
☐ ACAT III

15. Which phase of the Acquisition process is the program currently in?
- ☐ Concept Refinement
- ☐ Technology Development
- ☐ System Development and Demonstration
- ☐ Production and Deployment
- ☐ Operations and Support

16. Which statement reflects the amount of software that is COTS in this program
- ☐ None
- ☐ very little
- ☐ some
- ☐ large percentage
- ☐ 100%

17. How much of the programs software is replacing existing government software?
- ☐ none
- ☐ 0-50%
- ☐ 50% or more
- ☐ Don't know

18. Where do you rate the complexity of you programs software on the following scale?

simple automation 1 2 3 4 5 6 7 8 embedded real

time(example: a form program) example: fly-by-wire flight controls)

19. Roughly what percentage of your program cost is for software development?
- ☐ 0 -25%
- ☐ 25 – 50%
- ☐ 50 – 75%
- ☐ 75 – 100%
- ☐ Don't know

20. Select on the scale the amount of user involvement in the entire software acquisition process of your current program.

No user involvement 1 2 3 4 5 6 7 8 9 User Involved in every aspect

21. How long has your current program been on going? (months only required if <1yr)

__________ years _________months

22. How many government personnel (military and civilian) are dedicated to the software portion of your program?__________

23. Do you consider your current program to be software intensive?

- ☐ Yes
- ☐ No

24. What category does your current program belong to?

- ☐ Weapon system
- ☐ Support system
- ☐ Other (specify) ______________________________

25. If you were or are a member of the selection team or are knowledgeable about the selection of your software contractor, rank order (1 – 6; with 1 being the most important and 6 the least important) the following attributes according to their importance in selecting the contractor chosen for your program:

_____Past performance

_____Proposed cost

_____Proposed schedule

_____Knowledge of legacy system

_____Language expertise

_____Capability Maturity Model (CMM) certification level

_____ Not a member of selection team, or not knowledgeable about

contractor selection

26. Has the software contractor completed previous systems of the same scale?
    - ☐ Yes
    - ☐ No
    - ☐ Don't know

27. Does the software contractor have proven experience producing software in the same language as your program?
    - ☐ Yes
    - ☐ No
    - ☐ Don't know

28. Is the software portion of the program a fixed price contract?
    - ☐ Yes
    - ☐ No
    - ☐ Don't know

29. Was government and contractor team training administered at program initiation?
    - ☐ Yes
    - ☐ No
    - ☐ Don't know

30. The software contractor is certified:
    - ☐ CMM Level I
    - ☐ CMM level II
    - ☐ CMM level III
    - ☐ CMM level IV
    - ☐ CMM level V
    - ☐ Don't know

31. Was government and contractor team training administered at program milestones?
    - ☐ Yes at all milestones
    - ☐ No at none of the milestones
    - ☐ Some milestones but not all
    - ☐ Don't know

32. Has your current program undergone independent expert reviews (IER)?
    - ☐ Yes
    - ☐ No
    - ☐ Don't know

33. If yes, how often were IER's conducted?
- ☐ Quarterly
- ☐ Biannually
- ☐ Annually
- ☐ At each milestone
- ☐ Other (specify)______________________

34. Is this consistent with the planned IER schedule?
- ☐ Yes
- ☐ More frequent than scheduled
- ☐ Less frequent than scheduled
- ☐ We have no IER schedule

35. Do you consider the software portion of your program to be complex?
- ☐ Yes
- ☐ No

36. Select a number on the scale that reflects the amount of effort used to reduce software complexity

No effort **1 2 3 4 5 6** Every possible effort **Don't know**

37. Select the range below that best indicates your programs schedule performance index (SPI)
- ☐ 0 - .25
- ☐ .25 - .50
- ☐ .50 - .75
- ☐ .75 – 1.0
- ☐ Don't know

38. Select the range below that best indicates your programs cost performance index (CPI):
- ☐ 0 - .25
- ☐ .25 - .50
- ☐ .50 - .75
- ☐ .75 – 1.0
- ☐ Don't know

39. Is any portion of your software commercial off the shelf (COTS)?

- ☐ Yes COTS software used
- ☐ Considered COTS for use but chose not to use it
- ☐ No, never considered COTS
- ☐ Don't know

Program progress to date

40. There have been ______ modifications to the software contract as a result of changes or enhancements?
    - ☐ No
    - ☐ a few
    - ☐ slightly more than a few
    - ☐ many
    - ☐ Don't know

41. Approximately how many of the contract modifications were after initial testing?
    - ☐ None
    - ☐ less than half
    - ☐ more than half
    - ☐ All
    - ☐ Don't know

42. Since the initial acquisition program baseline (APB) there have been ______ changes to the software requirements?
    - ☐ no
    - ☐ less than 5
    - ☐ less than 10
    - ☐ less than 25
    - ☐ less than 50
    - ☐ less than 75
    - ☐ less than 100
    - ☐ more than 100
    - ☐ Don't know

43. Approximately how many of the software requirements changes were made after initial testing?
    - ☐ none, all were made prior to testing
    - ☐ less than half
    - ☐ less than ¾
    - ☐ nearly all
    - ☐ all
    - ☐ Don't know

44. How much has your program schedule changed due to software?
    - ☐ Not at all
    - ☐ Slightly
    - ☐ Significantly
    - ☐ Don't know

45. At any time were requirements removed to adjust for software requirements?
    - ☐ Yes
    - ☐ No
    - ☐ Don't know

46. How many software specific engineering change orders have been executed on your program?

- ☐ None
- ☐ a few
- ☐ Many
- ☐ Don't know

47. Does your program follow an iterative development process?

- ☐ Yes
- ☐ No
- ☐ Don't know

48. Software defect/trouble reporting rates are:

- ☐ Lower than expected
- ☐ As expected
- ☐ Higher than expected
- ☐ Don't know

49. How many times has your program been rebaselined since the initial APB?

- ☐ Never
- ☐ 1 – 5 times
- ☐ 5 – 10 times
- ☐ More than 10 times
- ☐ Don't know

Answer the following questions with regard to <u>all</u> programs in your experience, past and present.

Previous Program(s) Information

50. Have you ever been a member of an independent expert review on another program?

- ☐ Yes
- ☐ No

Statement	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
Acquisition Training					
51. Acquisition personnel are sufficiently trained in the management of software					
52. Software specific training is available for all acquisition personnel.					
53. There are enough AF acquisition personnel with software management skills					
54. The Air Force is effective at retaining key researchers					
55. In my current position there is time for professional development (i.e. professional reading)					
56. Professional reading is encouraged to advance my knowledge in the latest advances in software management.					
57. In my experience acquisition personnel take advantage of available software specific training.					
Contracting					
58. The Air Force uses the right criteria for selecting software contractors					
59. The most important criteria for selecting software contractors is cost					
60. The most important criteria for selecting software contractors is schedule					
61. The most important criteria for selecting software contractors is past performance					
Policy and procedures					
62. Software acquisition lessons learned are well documented					
63. In the past 5 years we have changed the way we manage software					

acquisition					
64. Software acquisition is performed the same way it has been since the AF began acquiring software					
65. We maintain sufficient software metrics on acquisition programs					
66. Metrics collected on software acquisition programs are analyzed					
67. Configuration Control boards are a crucial component of software program success.					
68. Air Force Acquisition programs use Configuration Control boards properly.					
69. Metrics are used in the software acquisition decision making process					

70. Please provide any additional comments that you feel would be helpful in this study:

____
____
____
____
____
____
____
____
____
____
____
____

Appendix B – Data Coding

Table 8 - Program Characteristics data coding (1)

	Question #	Measure	Question	Original answer format	Score Given
Scale	14	ACAT Level		1 = ACAT I	1
				2 = ACAT IA	1
				3 = ACAT IAM	1
				4 = ACAT IAC	1
				5 = ACAT II	2
				6 = ACAT III	3
	22	Size of Acquisition Team	How many government personnel (military and civilian) are dedicated to the software portion of your system?	Response given = Number of military and gov't personnel on software portion of program	Respons es used as is
	19	Cost	Roughly what percentage of your program cost is for software development?	1 = 0 - 25%	1
				2 = 25 - 50%	2
				3 = 50 - 75%	3
				4 = 75 - 100%	4
	23	Software intensity	Do you consider your program to be software intensive?	1 = Yes	1
				2 = No	0
	18	Complexity	Where do you rate the complexity of your program's software on the following scale?	1 = simple automation	1
				2	2
				3	3
				4	4
				5	5
				6	6
				7	7
				8	8
				9 = embedded real time	9

Table 9 - Program characteristics data coding (2)

	Question #	Measure	Question	Original answer format	Score Given
Support or Weapons	24	Support or weapons system	What category does your current program belong to?	1 = Weapon	1
				2 = Support	2
				3 = Other	3
Lifecycle Phase	15	Phase	Which phase of the acquisition process is the program currently in?	1 = Concept Refinement	1
				2 = Technology Development	2
				3 = System Development and Demonstration	3
				4 = Production and Deployment	4
				5 = Operations and Support	5
	21	Time	How long has your current program been ongoing?	A = Value for years	Combined to represent year value
			A = Years	B = Value for months	A + B/12
			B = Months		
Modernization or New	17	Modernization or New	How much of the programs software is replacing existing government software?	1 = None	1
				2 = 0 - 50%	2
				3 = 50 - 100%	3

Table 10 - Implement past recommendations data coding (1)

	Question #	Measure	Question	Original answer format	Score Given
Stress contractor past performance	30	CMM Level	The software contractor is certified:	1 = CMM Level I	1
				2 = CMM Level II	2
				3 = CMM Level III	3
				4 = CMM Level IV	4
				5 = CMM Level V	5
				6 = Don't Know	0
	25	Experience	Rank order contractor attributes as used in the selection of software contractor	Past performance was a selection to be ranked	If past performa nce was ranked:
					1 = 3
					2 = 2
					3 = 1
					else = 0
	26	Experience	Has software contractor completed previous systems of the same scale	1 = Yes	1
				2 = No	0
				3 = Don't know	-
	27	Experience	Does the software contractor have proven experience producing software in the same language as your program?	1 = Yes	1
				2 = No	0
				3 = Don't know	-

Table 11 - Implement past recommendations (2)

	Question #	Measure	Question	Original format	Score Given
Initiate Independent Expert Review	32	Occurrence	Has your current program undergone independent expert review?	1 = Yes	1
				2 = No	0
				3 = Don't know	-
	33	Frequency	If yes (Q32), How often were IER's conducted?	1 = Quarterly	
				2 = Biannually	
				3 = Annually	
				4 = At each Milestone	
				5 = Other (specify)	
	34	Planned for	Is this consistent with the planned IER schedule?	1 = Yes	1
				2 = More frequent than scheduled	1
				3 = Less frequent than scheduled	0
				4 = Program does not have IER schedule	0
Improve software skills of acquisition personnel	1	Experience	How many years experience do you have working in an acquisition position/function?	A= Years	A & B combined to provide one year score equal to:
				B = Months	A + B/12
	2	Experience	Approximately how long have you been a member of your current program acquisition team?	A= Years	A & B combined to provide one year score equal to:
				B = Months	A + B/12

Table 12 - Implement past recommendations data coding (3)

	Question #	Measure	Question	Original format	Score Given
Improve software skills of acquisition personnel	4	Experienc e	How many different programs have you been a member of the acquisition team for (counting you current program?)	Response = #	No change
	5	Experienc e	For each item below, indicate on the corresponding line a number from the scale that indicates your level of expertise in that area:	1 = Novice to 6= expert	
			Software		
			Hardware		
			Acquisition		
			Contracting		
			Program Management		
	6	College Education	Check all for which you hold an undergraduate degree	1 = N/A	Any degree = 1
				2 = Software Engineering	Software related degree = 2
				3 = Computer Science	
				4 = Information Resource Mgt	
				5 = Computer Information Science	
				6 = Business Mgt	
				7 = Electrical Engineering	
				8 = Other (specify)	

Table 13 - Implement past recommendations (4)

	Question #	Measure	Question	Original format	Score Given
Improve software skills of acquisition personnel (Con't)	8	College Education	Check all for which you hold an graduate degree	1 = N/A	Any degree = 1
				2 = Software Engineering	Software related degree = 2
				3 = Computer Science	
				4 = Information Resource Mgt	
				5 = Computer Information Science	
				6 = Business Mgt	
				7 = Electrical Engineering	
				8 = Other (specify)	
	10	Acquisition Training	Have you completed any software specific acquisition courses in the last 5 years?	1 = Yes	1
				2 = No	0
	10A		Number of courses completed	Number	No change

Table 14 - Implementation of past recommendations data coding (5)

	Question #	Measure	Question	Original answer format	Score Given
Improve software skills of acquisition personnel (Con't)	11	Acquisition Training	Check by any of the following DAU courses you have completed:	1 = IRM 101	1 for each attended
				2 = SAM 101	SAM weighted X3
				3 = ACQ 201	IRM weighted X2
				4 = IRM 201	
				5 = SAM 201	
				6 = SAM 301	
				7 = IRM 303	
	13	Acquisition Training	Check by any of the following certifications you currently hold	1 = contracting level I	1 for each attended
				2 = contracting level II	Information
				3 = contracting level III	Technology
				4 = Information Technology I	weighted X2
				5 = Information Technology II	
				6 = Information Technology III	

Table 15 - Implement past recommendations data coding (6)

	Question #	Measure	Question	Original answer format	Score Given
Improve software skills of acquisition personnel (Con't)	7	Software training	How many software courses did you complete in the pursuit of an undergraduate degree?	1 = None	No change
				2 = 1 - 2	
				3 = 2 - 3	
				4 = 3 - 4	
				5 = 4+	
	9	Software training	How many software courses did you complete in the pursuit of a graduate degree?	1 = None	No change
				2 = 1 - 2	
				3 = 2 - 3	
				4 = 3 - 4	
				5 = 4+	
	12	Software training	Have you completed any other software related courses? (non-degree seeking, certificate, certification, etc)?	1 = Yes	1
				2 = No	0
			If yes how many?	Number	
Collect Disseminate and employ best practices	35	Reduce Complexity	Do you consider the software portion of your program to be complex?	1 = Yes	1
				2 = No	0

Table 16 - Implement past recommendations data coding (7)

	Question #	Measure	Question	Original answer format	Score Given
Collect Disseminate and employ best practices (Con't)	36	Reduce Complexity	Select a number on the scale that reflects the amount of effort used to reduce complexity?	1 = No effort	If 1 - 6 No change
				2	
				3	
				4	
				5	
				6 = Every possible effort	
				7 = Don't know	-
	39	COTS used	Is any portion of your software commercial off the shelf (COTS)?	1 = Yes COTS software used	2
				2 = Considered COTS for use but chose not to use it	1
				3 = No, never considered COTS	0
				4 = Don't know	-
	29	Team Training	Was government and contractor team training administered at program initiation?	1 = Yes	1
				2 = No	0
				3 = Don't know	-

Table 17 - Implement past performance data coding (8)

	Question #	Measure	Question	Original answer format	Score Given
Collect Disseminate and employ best practices (Con't)	31	Team Training	Was government and contractor team training administered at program milestones?	1 = Yes	1
				2 = No	0
				3 = Don't know	-
	47	Iterative Developme nt	Does your program follow an iterative development process?	1 = yes	1
				2 = No	0
				3 = Don't know	-
Restructure Contract Incentives	28	Contract	Is the software portion of the program a fixed price contract?	1 = yes	1
				2 = No	0
				3 = Don't know	-
User Involvement	20	User member of Acquisition team	Select on the scale the amount of user involvement in the entire software acquisition process	1 = No user involvement	No change
				2	
				3	
				4	
				5	
				6	
				7	
				8	
				9 = User involved in every aspect	

Table 18 - Management success data coding (1)

	Question #	Measure	Question	Original answer format	Score Given
Frequency of rebaselining	49	Rebaselining	How many times has your program been rebaslined since initial APB?	1 = Never	4
				2 = 1 - 5 times	3
				3 = 5 - 10 times	2
				4 = more than 10 times	1
				5 = Don't know	-
Schedule slips	44	Schedule change	How much has your program schedule changed due to software?	1 = Not at all	3
				2 = Slightly	2
				3 = Significantly	1
				4 = Don't know	-
	37	SPI	Select the range below that best indicates your programs schedule performance index (SPI)	1 = 0 - .25	4
				2 = .25 - .50	3
				3 = .50 - .75	2
				4 = .75 - 1.0	1
				5 = Don't know	-
Requirements Removed	45	Requirements removed to adjust for software	At any time were requirements removed to adjust for software requirements?	1 = Yes	1
				2 = No	0
				3 = Don't know	-

Table 19 - Management success data coding (2)

	Question #	Measure	Question	Original answer format	Score Given
Contract Modifications	40	Contract changes	There have been _______ modifications ot the software contract as a result of changes or enhacements	1 = No	4
				2 = a few	3
				3 = slightly more than a few	2
				4 = many	1
				5 = Don't know	-
	41	Timing	Approximately how many o fthe contract modifications were after initial testing?	1 = None	4
				2 = Less than half	3
				3 = More than half	2
				4 = All	1
				5 = Don't know	-
Software Budget	38	CPI	Select the range below that best indicates your programs schedule performance index (CPI)	1 = 0 - .25	4
				2 = .25 - .50	3
				3 = .50 - .75	2
				4 = .75 - 1.0	1
				5 = Don't know	-

Table 20 - Management success data coding (3)

	Question #	Measure	Question	Original answer format	Score Given
Requirements Changed	42	Requirement changes	Since the initial acquisition program baseline there have been ______ changes to the s/w requirements.	1 = No	8
				2 = Less than 5	7
				3 = Less than 10	6
				4 = Less than 25	5
				5 = Less than 50	4
				6 = Less than 75	3
				7 = Less than 100	2
				8 = More than 100	1
				9 = Don't know	-
	43	Timing	Approximately how many of the software requirements were after initial testing?	1 = None	5
				2 = Less than half	4
				3 = Less than 3/4	3
				4 = Nearly all	2
				5 = all	1
				6 = Don't know	-
Engineering Change Orders	46	ECO's executed	How many software specific engineering change orders have been executed?	1 = None	3
				2 = a few	2
				3 = many	1
				4 = Don't know	-
Defect/trouble reporting rates	48		Software/defect trouble reporting rates are:	1 = Lower than expected	3
				2 = As expected	2
				3 = Higher than expected	1
				4 = Don't know	-

Appendix C – Survey Comments

The following seven comments were provided by survey respondents for survey item 70 which stated "Please provide any additional comments that you feel would be helpful in this study:"

1. To much emphasis is placed on the capability maturity model and level of maturity certification. My experience is each program may have the capability to perform as a high level, however, due to costs and/or program constraints, their corporate process is tailor such that they may be performing at a CMM level 2 vs. level 4. The emphasis should focus on the process being used on a given program, not the corporate level that a contractor is capable of performing at. The lightening bolt initiatives have removed the "big hammer" previously available to the government to manage programs. The pendulum needs to swing a little more towards the left, hence, giving the government team increased visibility, leverage and ability to manage large software efforts.

2. This is obviously "software centric." I don't have a "software program." I have a program that is software intensive. I don't have software requirements; I have system performance requirements. How much of the effort is dedicated to software is a WAG, as is number of software changes. Software is often changed to correct/mask a system problem. That doesn't mean the software was the problem to start with. I've tried to answer these questions as best I could, but I'm not sure I hit the mark.

3. If I am not mistaken, SPI and CPI can be greater than 1. I had a comment in mind for question 30, but I have forgotten what my concern was. If I recall correctly, there should be more choices for a response to that question. Additionally, I think I am more to blame

for not making time to obtain continuing education and training as well as advanced degree courses. Overall, however, it seems this survey should produce meaningful data.

4. The contract vehicle has let the fox rule the chicken coop and we just watch.

5. As I am fairly new to the AF and the acquisition process, I can only give you my two year experience. From what I have seen, by actively engaging myself in the program, software acquisition is very difficult to manage. Many of the program managers at the working level are Lts or Capts will little experience in software. These project officers make very important decisions on a daily basis that affect the program in the long term. Also, many of these project officers do not have engineering backgrounds, and many times are fulfilling a career broadening tour, which in turn creates uninformed or inexperienced decisions. With officers PCSing every 3 years, and programs lasting 5 or more years, it is very difficult to correct the mistakes of others. Don't get me wrong, I believe each officer does the best to his/her ability, but when bad decisions are made at the early stages of a program, then another officer takes over years later, it is difficult to have a stable program, especially when a significant amount of software exists. We have a contractor help us keep an eye on each program, knowing that many of them will be present throughout the life of the program, but these contractors can only inform us of the past, and advise us within our daily decisions. This many times creates a problem if the contractor is not fully engaged in the program. There are no easy answers to these problems, so I believe the first response to a successful program with large amounts of software is proper training of each officer, with a smooth transition of duties when an officer PCSs.

6. Many of the questions in this survey are not appropriate for software acquisitions in a spiral development program. More questions should be asked about how we do spiral acquisitions to prevent changing baselines and modifying requirements during development and test. We incorporate new requirements in a follow-on spiral to maintain schedule and budget for current work. That is the most effective way to control change and add capabilities to a software-intensive acquisition.

7. In the space acquisition arena, the most important criterion in selecting a software contractor is past performance; in other words, pick someone who knows what they are doing. The AF can fix software contractor management problems, but the AF does not have the expertise to replace a software contractor who doesn't have a clue as to how to do the work. Selection of the right software contractor will also alleviate future schedule and cost problems, too!

Bibliography

Cornella-Dorda, Santiago and Kurt Wallnau, Robert C. Seacord, John Robert (2000). "A Survey of Legacy System Modernization Approaches" CarnegieMellon software Engineering Institute, Technical Note CMU/SEI-2000-TN-003

Defense Acquisition University Course Catalog (2006), Defense Acquisition University, Fort Belvoir VA, accessed online at http://www.dau.mil/catalog/

Federal Register, Presidential Documents, Vol. 61 No. 140, Friday, July 19, 1996. Title 3 Executive Order 13011 of July 16, 1996, pp (37657 – 37662).

Mantel, Samuel J. Jr. and others. *Core Concepts: Project Management in Practice* (2nd Edition). Hoboken NJ: John J. Wiley & Sons, 2005.

Matthews, David F. COL (USA, Ret.) and Dr. Mark E. Nissen, Software Acquisition lessons learned through Student Research, May-Jun 2000, Army AL&T, p. 23-24.

McAlister, Paul, Historian, Air Combat Command (ACC), personal correspondence. 4 Aug 2005

Office of the Under Secretary of Defense for Acquisition and Technology. *Defense Science Board Task Force on Defense Software*, Washington DC, Nov 2000.

Office of the Under Secretary of Defense for Acquisition and Technology. *Defense Science Board Task Force on Acquiring Defense Software Commercially*, Washington DC, June 1994

Rosenthal, Robert and Ralph L. Rosnow. (1991) Essentials of Behavioral Research: Methods and Data Analysis; Second Addition. McGraw Hill, Boston MA

Sambur, Marvin R. Assistant Secretary of the Air Force (Acquisition). *Revitalizing the Software Aspects of Systems Engineering*, Memorandum, Sep 20, 2004. Washington D.C.

Schwab, Donald P. (2005) Research Methods for Organizational Studies; Lawrence Erlbaum Associates Inc., Mahwah New Jersey

Shore, L. M., Cleveland, J. N., & Goldberg, C. (2003). Work Attitudes and Decisions as a Function of Manager Age and Employee Age. Journal of Applied Psychology, 88: 529-537.

The Standish Group international (2001), Extreme Chaos, Article. http://www.standishgroup.com/sample_research/PDFpages/extreme_chaos.pdf

United States General Accounting Office. *Defense Acquisitions, Stronger Management Practices are needed to improve DoD's Software-Intensive Weapon Acquisitions*, Report to the Committee on Armed Services, U.S. Senate, March 2004

"U.S. Air Force AN/FYQ-93 Region Operations Control Center for Joint Surveillance System." Article. http://radomes.org/museum/equip/fyq-93.html . 7 Feb 2004

Acronym List

ACAT Acquisition Category

CMM Capability Maturity Model

CPI Cost Performance Index

DAU Defense Acquisition University

DSB Defense Science Board

GAO General Accountability Office

IRM Information Systems Acquisition Management (DAU course code)

SAM Software Acquisition Management (DAU course code)

SPI Schedule Performance Index

SPSS Statistical Package for the Social Sciences

Vita

Captain Dyan McClamma enlisted in the Air Force in 1987 and served as an electronic switching system and computer maintenance technician until 1998 when she graduated from Troy State University with a BS degree in Computer Information Systems. She received her commission upon completion of officer training school in September of 1998. Her first assignment as a communications officer was as AWACS software project manager, 552 Computer Systems Squadron, Tinker AFB, Oklahoma. While at Tinker she also served as the Computer System Group executive officer. Following this assignment Capt McClamma was assigned to the Joint Intelligence Center Pacific (JICPAC) Pearl Harbor, Hawaii and served as OIC of Systems Operations. It is from this assignment that Capt McClamma arrived at AFIT. After graduation Capt McClamma will remain at Wright-Patterson to serve at the National Air and Space Intelligence Center (NASIC).

Capt McClamma is married and the mother of three sons and one daughter.

REPORT DOCUMENTATION PAGE							*Form Approved* *OMB No. 074-0188*
The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of the collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to an penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**							
1. REPORT DATE *(DD-MM-YYYY)* 23-03-2006			**2. REPORT TYPE** Master's Thesis				**3. DATES COVERED** *(From – To)* March 2005 – March 2006
4. TITLE AND SUBTITLE Roadblocks to Software Modernization							**5a. CONTRACT NUMBER**
							5b. GRANT NUMBER
							5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) McClamma, Dyan E., Captain, USAF							**5d. PROJECT NUMBER** If funded, enter ENR #
							5e. TASK NUMBER
							5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way, Building 640 WPAFB OH 45433-8865							**8. PERFORMING ORGANIZATION REPORT NUMBER** AFIT/GRD/ENV/06M-09
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Lt Col Timothy J. Halloran 2950 Hobson Way, Building 640 WPAFB OH 45433-8865 937-255-3636 X7255							**10. SPONSOR/MONITOR'S ACRONYM(S)**
							11. SPONSOR/MONITOR'S REPORT NUMBER(S)
12. DISTRIBUTION/AVAILABILITY STATEMENT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.							
13. SUPPLEMENTARY NOTES							

14. ABSTRACT

Failed or troubled modernization efforts, such as the multi-million dollar 1997–2000 ROCC/SOCC failure, are a serious acquisition problem for the Air Force. Using both historical data and a survey of current Air Force software acquisition program key staff, this research examined the Air Forces ability to modernize legacy software systems.

The search of historical program data, to identify trends or similarities between known failed software modernization efforts, failed to uncover sufficient data for analysis. This lack of project data indicates a knowledge management issue (i.e. lessons learned are not recorded and stored so that they can be accessed by other programs) in the acquisition community.

The Phase II survey gathered data on current software programs and addressed the recommendations of the 2000 Defense Science Board (DSB) Study on Software. The goal was to determine first, had the recommendations been implemented, second, did program characteristics effect implementation, and third, did implementing the recommendations lead to program success. The survey results indicate that most of the recommendations of the DSB are not in practice in the acquisition community. They also indicate that support programs are more likely to have implemented the recommendations than are weapons systems.

15. SUBJECT TERMS
Software, Software Modernization

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Michael T. Rehg, PhD
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** *(Include area code)* (937) 255-3636, ext 4574 (michael.rehg@afit.edu)
U	U	U	UU	111	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18