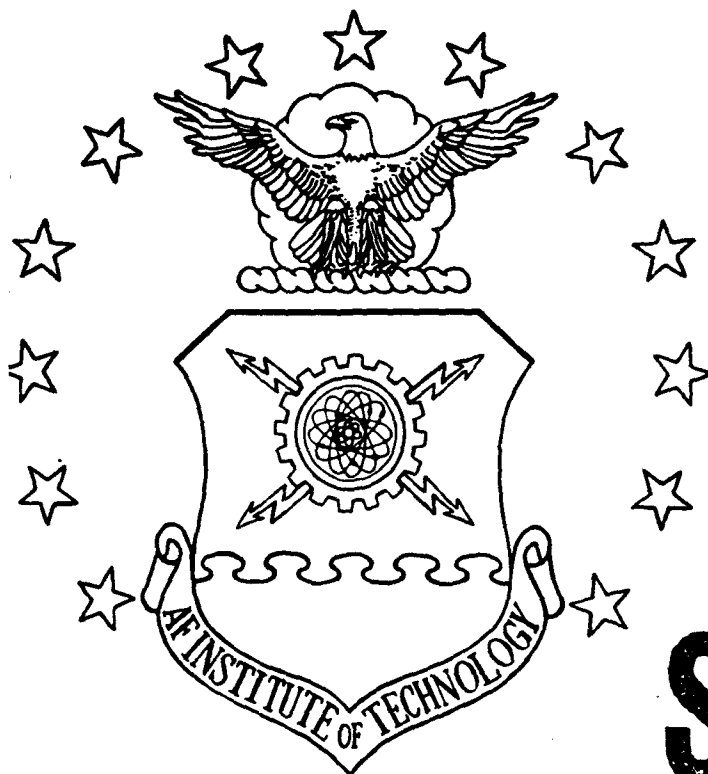


AD-A246 685



DTIC
ELECTE
MAR 02 1992
S D D

SELECTION OF A DEVELOPMENT
METHODOLOGY FOR THE ACQUISITION
OF COMMAND, CONTROL AND
COMMUNICATION SYSTEMS

THESIS

Martin W. Wituszynski, Captain, USAF

AFIT/GSM/LSY/91S-26

This document has been approved
for public release and sale; its
distribution is unlimited.

92-04903



DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

92 2 25 084

1

DTIC
ELECTE
MAR 02 1992
S D D

SELECTION OF A DEVELOPMENT
METHODOLOGY FOR THE ACQUISITION
OF COMMAND, CONTROL AND
COMMUNICATION SYSTEMS

THESIS

Martin W. Wituszynski, Captain, USAF

AFIT/GSM/LSY/91S-26

The views expressed in this thesis are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

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

AFIT/GSM/LSY/91S-26

SELECTION OF A DEVELOPMENT METHODOLOGY FOR THE
ACQUISITION OF COMMAND, CONTROL AND COMMUNICATION SYSTEMS

THESIS

Presented to the Faculty of the School of Systems and Logistics
of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Systems Management

Martin W. Wituszynski, B.S.

Captain, USAF

September 1991

Preface

The purpose of this study was to develop a method for selecting a development methodology for Command and Control programs. The idea for this research came about after discussions with the Strategic Command, Control and Communications System Program Office. Case studies were conducted on four of their current programs to see if certain program characteristics made either the conventional or incremental development methodology more appropriate. The cases did highlight some characteristics that should be considered prior to the selection of a methodology. Future research should be done to include additional development methodologies that were not covered due to time constraints.

In conducting the case studies and writing this thesis I have had a lot of help from others. I would like to thank my thesis advisor, Major Roger Koble, and reader, Mr. Dan Ferens, for their patience and guidance over the past year. I would also like to thank Lt Col Denner from the program office for his assistance and cooperation in conducting the case studies.

Martin W. Wituszynski

Table of Contents

	Page
Preface	ii
List of Tables	v
Abstract	vi
I. Problem Statement	1
General Issue	1
Specific Problem	1
Investigative Questions	2
Research Scope	3
II. Literature Review	4
Introduction	4
Justification	4
Conventional Software Development	5
Incremental Acquisition	7
Software Development Methodology Selection	14
Development Difficulties	16
Summary	19
III. Methodology	21
Research Method	21
Justification	21
Limitations	23
Research Process	24
Interview Procedure	26
IV. Results	30
Case Descriptions	30
Rapid Execution and Combat Targeting (REACT)	30
Ground Wave Emergency Network (GWEN)	35
Strategic Mission Data Preparation System (SMDPS)	38
Dual Frequency MEECN Receiver (DFMR)	43

	Page
V. Analysis	45
Incremental Development	46
Requirements Definition	46
System Complexity	48
Subsystems	48
User Involvement	50
Previous Systems	51
Funding	51
Development Expertise	52
Summary	53
Conventional Methodology	55
Requirements Definition	55
System Complexity	56
Subsystems	57
User Involvement	58
Previous Systems	58
Funding	59
Development Experience	59
Summary	60
VI. Recommendations	62
Implications for Development Managers	62
Implications for Researchers	64
Future Research	66
Appendix A: Interview Guide	68
Appendix B: Transcribed Interviews	70
Bibliography	126
Vita	128

List of Tables

Table	Page
1. Characteristic Matrix	46
2. Selection Criteria	63

Abstract

This study investigated command and control projects to see if certain program characteristics help determine the appropriate development methodology. A literature search revealed no previous research in this area; but did identify at least two possible development methodologies, and some suggestions for key characteristics. Four case studies were conducted to study the effects of certain characteristics on the success of the development methodology employed; either conventional or incremental. Requirements definition, availability of a previous system, system complexity, number of subsystems, development experience, user involvement, and funding all appear to impact the appropriateness of the two methodologies investigated.

SELECTION OF A DEVELOPMENT METHODOLOGY FOR THE ACQUISITION OF COMMAND, CONTROL AND COMMUNICATION SYSTEMS

I. Problem Statement

General Issue

The United States Air Force is currently developing and procuring many complex and software-intensive systems. Studies of past acquisitions of command, control and communication systems by the Office of the Under Secretary of Defense Research and Engineering and the Armed Forces Communications and Electronic Association (AFCEA) have concluded that conventional acquisition techniques have led to unsatisfactory results in many cases (5:v). The systems investigated were highly-complex, software-intensive programs, and they require special development. The identification of these special requirements have created problems for project managers. The Strategic Command, Control and Communications System Program Office (SPO) has posed the question: How can we best manage the acquisition of command and control systems?

Specific Problem

Two of the possible alternatives for acquiring these systems are Conventional Acquisition, and another

development technique known as Incremental Acquisition. Conventional Acquisition is defined as the development of an entire system using a structured process known as the waterfall model. Incremental Acquisition is defined as the delivery of a well-defined core capability to the field as soon as possible, and development of a full capability through incremental upgrades. The incremental upgrades are done by repetition of the waterfall model. Therefore, the main difference between these two development methodologies is the number of repetitions of the development cycle for incremental acquisition. This research investigates the use of incremental acquisition, as opposed to the conventional method, to solve the problem stated above by the Strategic Command, Control and Communications SPO. The overall research question is: " How does a manager know when a particular development methodology is appropriate?"

Investigative Questions

The investigation consists of four case studies of command and control projects that have used either conventional or incremental acquisition techniques. The number four was determined as the most efficient way of conducting a comparison of conventional versus incremental acquisition, while at the same time studying the impact on ability to meet the project's baseline requirements. The following questions were included in the interview guide, to

help define characteristics for determining which acquisition technique is appropriate for a specific project.

1. What level of software development complexity makes incremental design more desirable?
2. What number of system interfaces determines that the system should evolve incrementally?
3. Which subsystems of the project can be fielded early to meet a user requirement?
4. What level of understanding of the system requirements is needed to impact the need for incremental development?
5. What level of user involvement in development makes incremental development possible?
6. What level of design detail would necessitate incremental development?
7. What program experience level influences the acquisition technique employed by the manager?
8. How does the existence of a previous system affect the decision for acquisition technique?
9. What other system characteristics help determine that the system should evolve incrementally?
10. What project funding level influences the decision for acquisition technique selected?

Research Scope

This research is limited to the use of either conventional or evolutionary acquisition techniques for the development of command and control programs. Command and control projects have been isolated since they tend to be large systems and involve a complex software development. The study suggests factors to determine the appropriate acquisition technique for a specific system.

II. Literature Review

Introduction

This review looks at existing literature pertaining to two acquisition techniques available for C3 programs, and their implications for managing the development of these systems. This review investigates two possible acquisition techniques available: the conventional method, and what is known as Incremental Acquisition. Literature reviewed includes Department of Defense standards and regulations, Defense System Management College pamphlets, a National Research Council report, and books and articles on software engineering management.

This review first looks at the conventional and incremental techniques available when planning the acquisition strategy for C3 programs. Then it describes each technique in detail. Lastly, it examines how to determine when one of the techniques may be more appropriate for acquiring an individual C3 system, by giving several examples of decision criteria.

Justification

Command, Control and Communication (C3) is a system that gives leaders the ability to assess the situation, to make the appropriate decision, and to disseminate the decision into the field. America's weapons have been

upgraded during the 1980s, but "this modernization must be accompanied by corresponding upgraded Command, Control, Communication and Intelligence (C3I) systems if we are to take advantage of advancing technology and satisfying requirements (14:183)." Most of the military's C3 systems were designed and built during the 1950s and 60s. Many are located on fixed sites or depend on ground communication networks, and have become vulnerable to attack. The Air Force needs to correct the existing deficiencies to create a more survivable system. Electronics Systems Division has been tasked with developing these new systems.

Conventional Software Development

Current government contracts are usually written requiring the contractor to perform software development in accordance with DOD-STD-2167A, Defense System Software Development. This standard lays out a waterfall schedule to be followed when developing software for the Department of Defense. This process consists of the following activities:

1. System Requirements Analysis/Design
2. Software Requirements Analysis
3. Preliminary Design
4. Detailed Design
5. Coding and Unit Testing
6. Component Integration and Testing
7. Configuration Item Testing
8. System Integration and Testing (7:9)

The first step, System Requirements Analysis, is one of the keys to the success of this software acquisition technique. The developers need to sit down with the user and thoroughly understand the operational concept and requirements of the entire system prior to moving into system design. This is known as Top-Down design, because the system requirements are the basis for determining the requirements for the subsystems involved.

Most of the steps outlined above are followed by a specific review which must be completed before proceeding to the next step. These reviews ensure that no unknowns are carried into the next phase of development. The contractor is required to develop a plan for conducting these activities, and the plan must be approved by the government's contracting agency. This standard ensures that the contractor is following the structured design. 2167A applies to the planning, development, acquisition, test, support and use of computer resources in: 1) systems

acquired under AFR 800-2; 2) systems undergoing modification under AFR 57-4; and 3) prototypes and demonstrations of advanced technologies that may be candidates for use in systems that will be acquired under AFR 800-2 (6:1).

As stated above, this standard is required for the majority of projects being developed for the Department of Defense. The process is similar to that used for the development of military hardware. The waterfall model is typically oriented towards a single delivery date for the entire system. All analysis and design are done in detail, before coding and testing begin.

In order for this conventional methodology to work, the system requirements must be known up front. The user will not get any hands on experience until the system has already been designed and built. Changes at this point in the program would be time consuming and costly; so, the job has to be done right the first time.

Incremental Acquisition

AFR 800-14 defines Incremental Acquisition as a tailored variation for developing computer software where the initial requirements definition phase is abbreviated, focusing on selected core functions. User experience with early deliveries of core functions is are used to help define requirements for additional functions that will be added incrementally. (6:26)

This strategy dictates that the system will not have full capability when initially deployed, but will evolve to full capability through incremental upgrades. Incremental Acquisition consists of first defining the general structure of the system; and then sequentially defining, funding, developing, testing, fielding, supporting and evaluating increments of the system. Incremental Acquisition may be used to procure a system expected to evolve during development within an approved plan to achieve full capability. An underlying factor in this technique is the need to field a well-defined core capability in response to a requirement. (5:v)

The increments are treated as individual projects, with their requirements defined as a result of both continuous feedback from the user and the desired application of new technology balanced against the constraints of time, performance and cost. Changes should be accumulated and issued in batches as part of the next scheduled System Specification (12,172). This is different from the usual upgraded versions of the same program, because each increment adds additional capabilities rather than just improving the existing system.

According to the Defense Systems Management College,
for Incremental Acquisition to work it requires:

1. A concise statement of system requirements
2. A general description of system functional capability
3. A flexible, well-planned overall architecture
4. A plan for incremental achievement of full capability
5. Early definition, funding, development, testing, fielding, supporting and evaluation of core capability
6. Sequential definition, funding, development, testing, fielding, support and evaluation of increments
7. Continual dialogue and feedback with the user (5:3)

By developing and testing a core capability, even if it does fail, the developer will discover the problem early. Losses can be minimized at an early stage, and resources can be devoted to finding a better solution. If a major problem is discovered after an early upgrade, it would be easier to pinpoint the cause and remedy the problem in a later increment. (8:9)

This change-upon-learning process delivers a real capability to the user as early as possible. The added-value to the user can then be measured, and the design and objectives can be adjusted based upon observed realities. Early increments can be used operationally; and changes can be made to the existing system. Deming's 'eternal cycle'

looks at the benefits of modifying some existing system rather than starting from scratch. It is less risky to modify an existing system and easier to get end-user testing carried out, because the earlier increments are already being used operationally. (8:84)

The incremental approach provides a system to build upon for future increments. Tom Gilb sees several advantages of building on an existing system :

1. Improves existing system in shorter time frame
2. Does not disturb user with totally new (bug-ridden) system
3. Saves direct cost of system creation
4. Requires less overhead to study old system
5. Concentrates on solving pressing problems (8:227)

The incremental deliveries help the developer and the user understand the system better, and thereby produce a better end product. Incremental development is based upon iteration towards clear and measurable objectives. The increments are selected by determining the capability with the highest user-value to development cost ratio.

Incremental Acquisition can help the developer to understand and control the complex system by the means of one of the oldest management strategies, divide and conquer. User requirements may be altered as they get experience with the system. If new ideas are better than the original plan,

then developers must find practical and economic ways of implementing them as soon as possible. (8:90)

The incremental delivery of the system also has benefits to the development manager. The advantages of early results, according to Tom Gilb, include:

1. User is more supportive and cooperative
2. Subordinates are confronted with reality earlier
3. Team is exposed as incompetent if true
4. Obtain cost experience for later increments
5. Activity gains political credibility
6. Demonstrate capability early (8:112)

Incremental development can also help the manager program resources for later increments. Any project can be decomposed into a hierarchy of discrete events which describe the tasks to be accomplished. Task decomposition provides specific and verifiable measures of successful completion. It is especially beneficial because, for the first time, it becomes possible to predict at what time specific functional capabilities should be ready for demonstration to an often impatient customer. With the system broken up into increments, the manager can get a better feel for the accomplishments to date, and obtain cost/schedule data for future increments. (16:85)

Incremental deliveries will aid in the implementation of engineering changes. Changes written against a working system will have a larger probability of being operationally

satisfactory than those written against an abstract concept like an operational requirement. During implementation it is desirable that the number of changes are minimized. They are a distraction to the development and a major factor in project slippage. The user will appreciate getting something working and then applying changes to later increments. (15:96)

Sometimes, even with incremental development, there may be problems. Before the system goes operational, Stan Price describes a danger concerning changing user requirements.

Although the users who specified the system had current operational knowledge and a close dialogue was kept with future actual users during the life of the project, the passage of time could well mean that the original operational requirement which the system performs could be different from the current operational requirement. The system should therefore be designed from the onset for ease of change, with a modular structure and no built-in parameters. (15:59)

The subsystems still need to be designed for ease of change. Subsystems can be coded to make design changes simpler. One possibility, depth-first coding, involves coding all modules required to implement, test and demonstrate one program function at a time. This technique is favored by real-world developers who are constantly under pressure to deliver something that works. (16:148)

It is vital that all elements of the system are tested at all stages of integration and assembly from component and program modules. This will increase the probability that defects are detected as early as possible in the implementation process, when their correction can be most easily accommodated, rather than later in the process, when the higher level of integration is achieved, making their detection and solution much more difficult. Thus defects are less likely to cause delays and budget overruns. (15:101)

As pointed out by the literature available, Incremental Acquisition has several advantages when working with a loosely-defined, complex system. The incremental deliveries give the user hands-on experience in order to better define the overall system. The reason for multiple releases is that some systems are so technically difficult and so enormous in scope that they can not be finished in a single development cycle (12,172). This development technique also allows the developers to find errors earlier in the project, and thereby implement changes when it is more economic to do so. The increments provide an opportunity to go back to the analysis phase to ensure that the specifications are correctly understood and to confirm that the user's requirements have not changed (20,37).

Not all projects should be developed incrementally. The cost of repetition of the development life cycle has to be taken into account. The resources required may not be efficiently used if the project does not lend itself to this type of development.

Software Development Methodology Selection

Given the track record of most development projects, managers and users have been looking for a better way to develop software. Alternative development methodologies have been created to give developers different options. Now they must learn how to select the appropriate methodology for their project. Several examples of decision criteria are available for the selection of a development methodology.

Lee L. Gremillion and Philip Pyburn outline one selection method. Their method consists of "evaluating projects by the criteria of commonality, impact, and structure to help managers choose the appropriate development strategy and get applications to users faster (10:130)." Commonality is the probability that other organizations have a similar system. Impact is defined as the degree to which the system will affect the organization. Structure is how well the system requirements and the system itself are understood. The authors provide a matrix for selecting the best development method based upon these three properties.

Edward Yourdon explains his method for deciding whether to adapt a radical or conservative approach by answering the following questions:

1. How fickle is the user?
2. What pressure are you under to produce immediate, tangible results?
3. What pressure are you under to produce accurate schedule, budget, and estimate of manpower and other resources?
4. What are the dangers of making a major technical blunder? (20,57)

A radical approach is good when something must be working at a specific date, and when the user's perception of what he wants is subject to change. The conservative approach tends to be used on larger projects in which large amounts of money are being spent, and for which careful analysis and design are required to prevent disaster (20,59).

One of the premises of Yourdon's book is that many conventional projects tend to be over budget, behind schedule, unreliable and unacceptable to users (20,1). The larger the scope and size of a project, the more likely it is to have these problems. A project involving 100,000 lines of code is sufficiently complex that the system requirements are not thoroughly understood, or the requirements may change during the 2-3 year development. For projects having over one million lines of code conventional techniques and a conventional project lifecycle are almost guaranteed to fail (20,4).

The National Research Council has determined that the waterfall model has yielded successes when there is:

1. a stable set of requirements that are not significantly different from a previous system.
2. a system architecture and design that will satisfy the requirements.
3. a development team that communicates effectively and have previous experience on a similar system.

It seems that the waterfall model is appropriate for precededented systems, where the requirements are understood and experienced personnel are available. (13,21)

The last method to be discussed was presented by Alan Davis, Edward Bersoff and Edward Comer. They feel that some project aspects will affect the choice of development methodology. Examples of these aspects might include 1) requirements volatility, 2) the shape of requirements volatility, 3) the longevity of the application, and 4) the availability of resources to develop or effect changes.

Development Difficulties

The Defense Systems Management College states that difficulties arise during acquisition of C3 systems because:

It is often not feasible to define in detail the operational capabilities and functional characteristics of the system. If development of the system is undertaken without a clear definition of the operational capabilities and functional characteristics, then it is very likely that the development process will be long, costly, and unstable, and that the system will be unsatisfactory. (5:1)

The same point of view is expressed by Frederick Brooks, who states that

much of present-day software-acquisition procedure rests upon the assumption that one can specify a satisfactory system in advance, get bids for its construction, have it built, and install it. I think this assumption is fundamentally wrong, and that many software-acquisition problems spring from that fallacy. (3:25)

Since the system level requirements are usually difficult to understand up front, Tom Gilb states that the system requirements phase should be modified for certain projects:

One of the great time-wasters in software projects is detailed requirements analysis, followed by detailed design, followed by coding and testing phases. If only we had the intellectual capacity, and the necessary knowledge to do those things accurately! In reality, we have to admit that we cannot tackle such tasks adequately for any but trivially small projects. There are too many unknowns, too many dynamic changes, and too complex a set of interrelationships in the systems we build. We must take a more humble approach. It is absurd to spend - in fact waste - so much time at the beginning of a project to speculate on requirements and technical design attributes which can be measured much more cheaply and reliably if it is done while we implement a real working system. (8:90-1)

Even with the benefits of Incremental Acquisition, there are still objections. These objections to Incremental Acquisition include:

1. Our system can not be divided into smaller parts
2. We are in a hurry
3. Management will not like it
4. Designers can not make evolutionary plans
5. It is not the traditional way
6. The extra effort between steps will cost more (8:108)

Despite the objections, Tom Gilb states that experience shows that incremental acquisition will be the surest way to deliver the most critical results much earlier, and that it will probably deliver the entire long-range solution earlier than you would have otherwise done. The extra cost for the additional increments does not exist in reality, because defects show up early and can be cleared up early. In conventional planning, too large a commitment has already been made to design ideas before there is any feedback.
(8:108)

Many of the benefits from incremental acquisition are due to a change in development philosophy. Conventional planning asks the question: "how much can we accomplish within constraints?" while Incremental Acquisition asks a very different question: "How little developmental resources can we spend and still accomplish something useful?"

William Roetzhiem feels that this difference in philosophy has lead to unsatisfactory results if applied incorrectly:

It has been my experience that up to half of all potential software projects are born losers. Some projects are not technically feasible, while others are possible, but not within the framework of time and money that the customer is willing to accept. Some projects are so vaguely defined that the software requirement cannot even be approximated. (16:8)

If the appropriate development methodology is not selected, difficulties are likely to occur.

Summary

The increasing cost and complexity of today's C3 systems require the DoD to acquire these systems efficiently and effectively. Studies of past acquisitions of C3 systems by OSD and the Armed Forces Communications and Electronics Association have concluded that use of conventional strategies have often lead to unsatisfactory results. The findings of those studies stress the point that consideration be given to incremental acquisition. Unique characteristics of each program should be considered, and the acquisition strategy chosen must be consistent with basic DoD acquisition policy. (5:v)

While DoD standards require the application of DOD-STD-2167A for software development, Government policy explicitly calls for tailoring an acquisition strategy to meet specific needs and circumstances of the program. The current policies can also be changed if found to be inappropriate.

The systems considered in the 'unsatisfactory' studies discussed above were large, software-dominated information systems intended to aid commanders in performing C3 functions. (5:2)

Successful execution of Incremental Acquisition requires changes to conventional acquisition. The system development programs would require closer interaction between the developer and the user. Streamlined procedures would be needed to allow each increment to progress rapidly through the acquisition process. It must also be understood that Incremental Acquisition is not a single strategy ready for application to all C3 system acquisition efforts. (5:4)

Literature was reviewed discussing methods for determining the appropriate development methodology for a project. Several sets of decision criteria were presented for selection of the development methodology. These were used as a starting point for this research. The remainder of this paper describes existing development projects and then proposes a method for determining whether conventional or incremental acquisition is appropriate for a given project.

III. Methodology

Research Method

In order to solve the research problem stated in chapter one, four case studies were conducted of selected command, control and communication programs currently being developed at Electronics Systems Division. The study was exploratory, since it attempts to identify project characteristics which affect the acquisition technique used. The purpose is to propose a method for determining the appropriate acquisition technique for command, control and communication programs given certain characteristics of the program.

Justification

The following factors can be used to determine what research method should be used:

1. the type of research question posed
2. the extent of investigator control over events
3. the focus on contemporary or historical events (19:13)

The research question posed is: "How does a manager know when a particular development methodology is more appropriate? This question covers a broad spectrum of command, control and communication projects. It would be impossible to study all of them, so sampling was required. The question is also difficult to handle using an

experimental design. The development lifecycle for a major weapon system is several years. This time frame does not allow for direct observation of the entire development period. Finally, manipulation of the control variables is not possible. It would not be economical to develop the same system using different methodologies unless it was an existing risk reduction effort by the Air Force. No such programs have been located to date. Yin proposes that:

In general, case studies are the preferred strategy when 'how' or 'why' questions are being posed, when the investigator has little control over events, and when the focus is on a contemporary phenomenon within some real-life context. (19:13)

The case study should be the best way to propose a method to determine the better acquisition technique considering the research question posed and the attributes of the weapon system development process. Izak Benbaset and others see the case study as useful for this type of research because:

1. The researcher can study information systems in a natural setting, learn about the state-of-the-art, and generate theories from practice.
2. The case method allows the researcher to answer 'how' and 'why' questions, that is, to understand the nature and complexity of the processes taking place.
3. A case approach is an appropriate way to research an area in which few previous studies have been carried out. (2:370)

Limitations

There are some limitations inherent to the Case Study methodology. Stone identifies several disadvantages of the case study:

1. It is the least systematic of all research strategies.
2. Causal inferences from case study data are impossible since there are no control over confounding variables.
3. Data collection may alter the setting under study.
4. Hypothesis testing is not possible using case study data.
5. Results of case studies are likely to have substantial amounts of bias because of non-systematic collection, condensation, and interpretation of data.
6. Generalization from a case study's findings is not possible.
7. Case studies are more time consuming than other strategies. (18:25)

Care was taken to ensure that a systematic approach was used. This was done by using the same interview guide for each case, collecting the same information from each case. This was done to also eliminate researcher bias, and to ensure that the settings were not altered. Since the case study is not an experimental design, the researcher had no control over external variables. This made it difficult to clearly define causal relationships, but program similarities helped cancel out external variables. The cases studied are only a sample of command, control and

communication projects. In order to make generalizations about all projects of this type it must be assumed that these projects are representative of the entire population. The cases were selected as being a good cross section of command and control programs. The fact that case studies are time consuming was unavoidable given this research problem.

Research Process

Data was collected by interviewing project managers, program control officers, systems engineers, and the users of existing command and control programs. Program documents were also used to support the information gathered from project personnel. This way data was collected consisting of project characteristics, acquisition technique, and progress as variables. Data on the following characteristics was collected: level of software development, number of system interfaces, usable subsystems available, knowledge of system requirements, user involvement, design definition, manning level, funding, and existence of a prior system. Data was also collected for any additional characteristics that impacted the decision for development methodology.

Four programs were identified; two using conventional acquisition techniques, and two using incremental acquisition. The programs are Strategic Mission Data

Planning System (SMDPS), Ground Wave Emergency Network (GWEN), REACT Communications Element (RCE), and Dual Frequency MEECN Receiver (DFMR). Two programs were needed for each group so that the impact of program characteristics on the ability to meet the program baseline could be studied. The determination of program progress included qualitative and quantitative measures of actual cost, schedule, and system performance criteria versus the program baseline. Departures from the program baseline were accumulated to categorize each program. Changes in program direction were taken into account when determining program progress.

The following steps were necessary to conduct my research:

1. Select programs by acquisition technique used
2. Determine if program met cost/schedule/performance baseline
3. Construct interview guide
4. Dry run guide for understanding/flow
5. Conduct Interviews
6. Identify relationships between program characteristics, methodology, and progress.
7. Propose a method to decide acquisition technique.

Interview Procedure

Lincoln and Guba identify a general procedure for conducting interviews:

1. Decide on whom to interview.
2. Prepare for the interview.
3. Opening
4. Interview
5. Closure (11:25)

Program managers, program control officers, and users were interviewed, because they were thought to have the information required to perform the analysis. The mode of interviewing for this research approximated normal conversation. The way the researcher probed for detail, clarity or explanation, and his gestures gave him the means for shaping the interview. The interviewer did not use a specific, ordered list of questions because such formality would have destroyed the conversational style. He had a list in mind or in hand, but he was sufficiently flexible to conduct the interview in a natural way. (17,73)

To maintain a condition of balance - where the interviewer did most of the leading and the respondent did most of the talking - the interviewer set the stage with a general statement preparing the respondent for what was to follow. Yin sees some skills that are required to conduct good interviews: 1) Ask good questions, 2) Be a good listener, 3) be adaptive and flexible, 4) Have a firm grasp

of the issues being studied, and 5) Be unbiased by preconceived notions. (19,56)

In order to ask good questions the investigative questions identified in chapter one were used as a starting point. The questions were reminders to the investigator regarding the information that needed to be gathered, and the main purpose of these questions was to keep the interview on track. (19,70) The interview was adapted for each respondent depending upon how his job impacted the information required. Data was also collected on the individual's feelings about the selected methodology.

To help the interviewer do a better job of listening, the interviews were conducted in person. This allowed the interviewer to be more personal and to observe any nonverbal responses. To really listen to the respondent the interviewer must understand the perspective of the person being interviewed, and hear what is said without any preconceived notions. A tape recorder was used to fully document the conversation without detracting from the discussion.

The interviewer adapted each interview to the situation at hand. The interview was left open enough to take advantage of new discoveries. A good grasp of the issues involved made it easier for the interviewer to interpret the information and to redirect the conversation as required.

System descriptions and Program Director's Assessment

Reports were obtained prior to conducting the interviews in order to aid in the researcher's understanding of the issues.

The informal quality of these interviews allowed for follow-up discussion. "Early interviews tend to be less economical than later ones, mainly because the researcher has not yet fully determined precisely what information he needs; also, he is not always certain of what the respondent is telling him." (17,71) Additional questions surfaced after talking to other people and learning more about the organizations. It was important that the content of each interview is the same so that comparisons could be made. The follow-up conversations were more specific and were done over the telephone.

There did not seem to be any unusual aspects of this method. However, there were some significant obstacles to overcome. An interview guide was created to help shape the conversations. Next, the interviews had to be conducted to generate the data, and to identify any relationships between the variables. The interviews were conducted in person unless additional information was required after the initial visit. Finally, the method for determination of the best acquisition technique given the program characteristics listed above, had to be constructed.

The analysis was performed by comparing the characteristics of the two projects for each development

methodology. A comparison was also done on the two projects that met their baseline requirements. The existence of differences found during the comparisons, and pattern matching were used to propose a decision method.

IV. Results

Case Descriptions

Before the analysis of the data collected during the interviews is presented, a description of each program will be given. The descriptions include an explanation of the development method used, and a discussion of other program characteristics.

Rapid Execution and Combat Targeting (REACT)

REACT consists of a modification to upgrade SAC's Minuteman and Peacekeeper in Minuteman Silos Launch Control Centers (LCCs). REACT is divided into two portions:

a. The first portion of REACT, the Higher Authority Communications/Rapid Message Processing (HAC/RMPE), includes HAC integration, LCC alarm integration, rapid message processing, and performing error correction on National Command Authority (NCA) messages. The ICBM C3 System Program Office is responsible for the acquisition management of the HAC/RMPE portion of the REACT program.

b. The second portion of REACT, the Weapon Systems Control Element (WSCE), includes processing and implementing NCA messages, performing rapid retargeting of missile forces, and integrating LCC REACT equipment into one console that facilitates missile combat crew member console operation. The WSCE portion is managed by the Ballistic

Missile Organization (BMO), and was not included in the case study.

REACT is being developed conventionally because this was perceived to be the quickest way to field the new system. The plan is to develop the entire system at the same time. The contractor has run into some technical problems because of the complexity of the task. The Program Director's Assessment Report highlights a memory reserve problem, and a significant cost variance. The contract requires a 50% memory reserve capability in the system to allow for future growth, but the current software design utilizes more memory than originally anticipated. The contractor could redesign the code or add additional memory to the computer, but the contractor would have to pay for that effort with their own funds. The fact that the contractor is already in a loss position complicates the situation. The contractor expended more money than they anticipated in the system requirements derivation phase, and is reluctant to spend more company funds if avoidable. This is not the best environment for program management to work with. The following issues have impacted the progress of the development program.

REACT, as explained by one project manager, "is the most significant ICBM modernization in 25 years. It's going to be like night and day." The program is going to automate a lot of the manual processing done today; giving the system

a rapid targeting capability. Under the HAC/RMPE portion, the effort consists of HAC integration, LCC alarm integration, rapid message processing, and error correction on NCA messages. At the same time, this portion has to be integrated into WSCE, which is being developed concurrently. This level of complexity has lead to a cost variance and memory reserve problem. According to the program control officer, "in order to come to grips with the requirements and basic design, they had to use people that were much more talented than they really wanted to. They had more expensive people, more of them, and worked them longer than they expected." The contractor has not been able to recover all of the money spent during the system requirements derivation phase. This resulted in a cost overrun.

The complexity of the requirements has also lead to the memory reserve problem. The project manager estimates that 70% of the program is dedicated to software. The software consists of 60,000-70,000 SLOC of very intricate code. The project manager also stated, "Maybe GTE wasn't smart enough. Maybe they didn't have a good enough model because they didn't have a strong understanding of all the requirements in the beginning." The lack of understanding is most likely due to system requirements definition problems.

One of the factors impacting the REACT development is the need to field a system quickly. This has driven many decisions in the program. The project manager says,

We didn't have the luxury of well thought out requirements done for us. Usually what happens is that a lot of your operational requirements analysis should be done before you go out and do the actual engineering phase. We did most of the requirements during the time when you should be driving out your lower level specifications.

The contractor had been forced to estimate the cost of the effort before the system level requirements were identified. The user is going to reexamine requirements as problems arise in the program.

The software developed for REACT is being programmed in Ada. There is not a lot of Ada experience available in the SPO, but the REACT SPO obtained a software manager with six years of experience in software acquisition. Recently, that person was reassigned to another program, and was replaced by somebody with very little acquisition experience. This has made a difficult development effort even more difficult.

Another factor making development riskier is the decision to concurrently develop and produce the system in order to field a capability more quickly. The project manager feels that by making a production decision before development testing, "If you hit and you're on the mark, you're ahead of the ball game. But, if you make a production decision and find out later that you have a lot of problems, then it puts you back behind the power curve." The developers have only one chance to make this methodology work.

The REACT integration effort is very involved. Not only does the program consist of the two parts managed by the different product divisions, but these two parts can also be broken down. The command and control portion includes HAC integration, LCC alarm integration, rapid message processing, and error correction on NCA messages. The WSCE portion includes processing and implementation of NCA messages, rapid retargeting, and console development. It was decided to develop all capabilities in parallel.

The user has been involved with the REACT program from within the organization. There are missile launch control officers located in the SPO as project managers. The project manager interviewed explained his communications with the users, "we call on a daily basis, because we're such a tight community." They are able to ensure that the user's needs are identified, and that validated requirements are incorporated. This has been a very important factor since the system level requirements were not defined ahead of time.

Ground Wave Emergency Network (GWEN)

The Air Force learned about the effects of an electromagnetic pulse (EMP) on communication networks during atmospheric testing at Johnson Island back in the '60s. In an EMP scenario, a nuclear weapon could be detonated that would charge particles in the ionosphere and render standard communications useless. The US Strategic Forces need a way to communicate after such an event. The Air Force thought that they could construct a system, using a ground hugging wave, to operate in that scenario.

The objective of GWEN is to increase the NCA and US strategic force's capability to maintain critical CONUS long range command and control communications despite physical damage and ionospheric disturbances caused by high altitude nuclear detonations. The equipment is designed to make GWEN a survivable, integrated communications network, and consists of unmanned relay nodes, input/output radio terminal equipment for injecting messages into and receiving messages from the GWEN system, and receive-only radio terminal equipment for reception of GWEN messages. The input/output terminals were installed at those organizations and installations directly involved in the command and control of US strategic forces. The total number of input/output terminals will be limited in order to maintain strict control over what messages are input to the system. Receive-only terminals are installed at SAC main operating

bases. Relay nodes will be located throughout the continental US, permitting diverse routing of a message if relay nodes are disrupted due to preventative maintenance or adverse actions. The relay nodes will be installed at ground wave radio ranges across the US.

The GWEN program is divided into three phases. Phase I is the Initial Conductivity Capability and was the concept validation phase. Phase II is the Thin Line Conductivity Capability, and is the GWEN prototype network. During Phase III, the Thin Line Conductivity Capability (TLCC) will be expanded into a Final Operational Capability.

This incremental development of the GWEN system has lead to satisfactory results. Phase 1, the initial connectivity capability, was developed to prove that the ground wave system actually worked. The concept was validated by tying a command post in with a few operating stations. It worked as planned; so the program office got the green light to proceed with Full Scale Development (FSD). The program went into Phase II, the TLCC. During this phase a prototype system was fielded to connect Headquarters SAC with its main operation bases around the US. The TLCC provided a limited number of paths for message transmission. Phase III, the Relay Node Network Expansion, is providing forty additional relay nodes into the system. The GWEN Maintenance Notification Center is being developed during this phase to inform relay nodes of the status of

their neighboring relays. This will help determine the best path for messages to travel during times of attack or maintenance. The Phase III equipment has already been developed. The only thing keeping this phase from proceeding on schedule is political concerns about the health implications of electromagnetic emissions from the system. The Phase III contract remains on cost and on schedule after two years with regard to contractor performance, but, due to proposed Congressional language in the FY91 Defense Authorization Act, the program is expected to experience a twelve month government delay at an estimated cost of \$14-16 million.

The program office realized the complexity of the GWEN project, and were willing to make a substantial investment. Therefore, they supported the development of a \$6.7 million software support facility.

The user had many opportunities to impact the systems requirements. The concept validation system was used by SAC during operational test and evaluation. Their comments were incorporated into the next phase. The Thin Line Connectivity Capability, the prototype GWEN system, was delivered to SAC giving them a chance to use the system operationally without having to wait for the expanded system. Assessments during that phase ensured that the system did what SAC had intended it to do. The prototype system has also lead to changes in the Maintenance

Notification Center, which has evolved into more than originally intended. Other changes include providing an integrated support facility and a software development facility.

Strategic Mission Data Preparation System (SMDPS)

The Phase I SMDPS effort started in 1979 to support B-52 Offensive Avionics System modified aircraft and the Air Launched Cruise Missile (ALCM). Phase I provides SAC with the ability to load mission data into the ALCMs. SMDPS Phase II provides the capability to prepare missions for the B-1B, B-52, and their associated weapon systems. The system also accepts mission data from the Headquarters SAC TRIAD Computer System (TRICOMS). The Phase II application software has been upgraded annually for system enhancements and to maintain/support compatibility with TRICOMS and evolving weapon systems.

SMDPS Phase III was supposed to provide an interactive planner to allow missions to be planned at the unit level, plus generate mission materials. Due to problems in the Phase III program, the program has been restructured and renamed the Nuclear Mission Planning and Production System (NMPPS). NMPPS currently consists of three blocks:

- a. Block I replaces the SMDPS Phase II Perkin Elmer equipment with modern, maintainable, commercial-off-the-shelf (COTS) NMPPS hardware suites at the SAC fixed

operational sites. The replacement will require the contractor to rehost the SMDPS Phase II software on the new hardware suite. There will be a deployable NMPPS Block 1 capability.

b. Block II will add the Combat Mission Folder capability. This will be a work station to generate crew mission materials automatically; a task that is currently done manually.

c. Block III is envisioned to provide SAC with their interactive mission planning system. This would allow crews to perform mission planning at the unit level.

The need for SMDPS came with the invention of aircraft avionics and smart weapons. Many smart weapons have their own navigation capabilities and are completely autonomous of the aircraft once launched. Because of these weapons and avionics systems, SAC aircraft are very data intensive. This data must be loaded into the aircraft and the missiles prior to launch. Instead of keying the information manually, SAC wanted an automated data preparation system.

The SMDPS program was organized to incrementally deliver a complete mission planning system. Each of the phases were planned to give the user additional capabilities. SMDPS ran into some problems using incremental development. SAC is currently using the Phase I and II systems operationally, but problems came when SAC was developing Phase III. In-house programmers tried to develop

an interactive system so that they could plan missions at the unit level, plus have the system automatically generate crew mission materials. According to the current program manager, after the first eighteen months of development, the program had already slipped by thirty four months. SAC turned to Systems Command for help. The program has now been restructured into what is called the Nuclear Mission Planning and Production System (NMPPS). NMPPS is currently in source selection to give the effort to a contractor in three blocks. Hopefully the restructured program will avoid some of the problems encountered in the earlier phase.

A mission planning system like this requires an extensive software development. Phase II SMDPS currently contains over 1.3 million lines of application software code. The program manager estimates that approximately 95% of the program effort is focused on software development. The general thrust in the DOD is to use Commercial off the Shelf (COTS) hardware whenever possible. The majority of the manpower for this effort is concentrated on developing the software required at the heart of the system.

Software experience was a big variable in the SMDPS program. As stated by the current program manager, "You can have the smartest programmer in the world and put him in as program manager in a software development effort, and he'll fail miserably." The user also had very poor requirements discipline because of the lack of management expertise.

With the programmers managing the development, requirements kept changing. This was because the programmers did not understand the need to baseline requirements. Without a baseline to design to, the program never progressed to actual software design.

The situation outlined above has been changed. Management responsibility has now been given to Systems Command. System Command is going to award the effort to a contractor. Also, the requirements definition has been formalized with the user. The program office now has a single point of contact, SAC/XR, for operational requirements.

The Mission Planning System is broken up into four subsystems: missile data preparation, aircraft mission planning, combat mission folder, and the interactive mission planner. The program manager felt that it would have been too much to develop them all concurrently. By developing them incrementally, it allows the government to get a product to the users to assess success and to make changes if needed.

The user has been involved in the development throughout the program. SAC was initially doing some of the work in-house in order to have control over the system requirements. Although that lead to problems, it did keep the user involved in the development. The user still has input to requirements on a regular basis . One of the ways

that the user impacts requirements is through the Material Improvement Program. Experience with early deliveries has lead to improvements in future deliveries.

The Mission Planning program manager stated that, "When you have a finite pile of money, the best way to get a realistic contractor response is to break the program up into smaller pieces." This allows you to deliver a capability to the field and assess success earlier. Incremental delivery also decreases program risk by allowing the contractor to level out manpower requirements. As work ends on one increment, personnel can be reassigned to the future increments. Even if funds are cut, some capabilities can still be delivered, and others can be shifted to a later increment when the funds are available. The program office is currently working on only the first two blocks, because resources are not yet available for Block III.

Dual Frequency MEECN Receiver (DFMR)

DFMR is a follow-on program to the Miniature Receive Terminal (MRT), which was built for the B-1 aircraft. When the Peacekeeper program asked for the same capability, ESD started a program to modify the MRT to meet the Peacekeeper/Rail Garrison requirements. SAC also wanted to replace some of their aging systems, so the design is being changed to also work in the Minutemen LCCs.

The purpose of the DFMR program is to take the existing MRT system, retune it to different frequencies, and add capabilities from the GWEN airborne platform. The MRT received on one frequency band, but had excess capacity to possibly handle an additional frequency. The receiver could be retuned, and incorporate GWEN capabilities, without developing a totally new receiver. It is basically an integration effort of technology already available in production systems. The DFMR consists of only two boxes, one of which was developed for a previous program. Some minor modifications may be required to make it interface with the second box; but the only real development is the second box. With the requirements identified up front, it was easier to lay out the system's development schedule. The contractor has remained close to that original plan.

DFMR is being developed using a traditional methodology. The conventional methodology seems to be working for the DFMR program. The only problems identified

in the Program Director's Assessment Report deal with contractual disputes, delivery of technical data, and a cost overrun. The reason for the cost overrun, according to the program manager, is "not because it's harder than anticipated, it's because we put him in a must-win situation." The contractor had lost on the MRT program and did not want to lose again, so they were very optimistic in their proposal in order to win the source selection. There are some schedule problems "because we asked them to do a 36 month program in 30 months." Except for the pressures of competition and fixed price contracts, the DFMR program seems to be on track.

The level of effort for software on the DFMR program consists of only about 10 thousand lines of code. The program office has contracted out its technical engineering management. For this program there are three software engineers, and six government employees managing the effort. The program manager feels that this manpower level is sufficient to develop the software for the receiver.

V. Analysis

Based on the results in the previous chapter, several program characteristics stand out. These characteristics help to distinguish one program from another. Characteristics were selected based upon observations made during a comparison of the cases. All of the cases were impacted by the following characteristics: requirements definition, system complexity, number of subsystems, user involvement, previous systems, funding and experience. The goal of this research is to construct a method for selecting the appropriate development methodology.

There seems to be a pattern between development methodology, program progress and the characteristics discussed. By comparing the characteristics of the two programs for each methodology and looking at the differences between them, this research can help demonstrate which characteristics are needed for a successful program. Table 1, which shows the relationships, helps to demonstrate this pattern.

	GWEN*	SMDPS	DFMR*	REACT
Methodology	Incr	Incr	Conv	Conv
Requirements	Yes	No	Yes	No
System Complexity	Yes	Yes	No	Yes
Subsystems	Many	Many	2	Many
User Involvement	Yes	Yes	Yes	Yes
Previous System	No	No	Yes	No
Funding Impact	Yes	Yes	No	No
Experience	Yes	No	Yes	No

* indicates successful program

Table 1 - Characteristic Matrix

Incremental Development

The following section will analyze the findings for the incremental cases. The research will attempt to show how these characteristics can impact the success of the program.

Requirements Definition. It appears that the system level requirements must be defined at the beginning of the program for this methodology to work. The requirements for the GWEN system were understood from the beginning of the program. The technology employed in this network is similar to that used in a radio station. The Air Force knew what the system would look like, but decided to incrementally deliver the system to develop the technology for command and control purposes. A structure was set up to systematically

deliver the final system. This was done by delivering a concept validation network, a prototype network, and the complete system in increments. These increments allowed the user to better understand what he needed to communicate in an EMP environment.

The requirements for the SMDPS were not so straightforward, and led to an eventual program slip. The development of smart weapons and aircraft offensive avionics made SAC's aircraft data intensive, because data is required for the navigation of both systems. The user wanted an automated system to load the data into these systems. The requirements were easier to define for the earlier phases, because the system was broken down into more manageable parts. The Phase III effort turned out to be more difficult. The present program manager said that there was no control over requirements. Since the programmers did not understand the need to baseline requirements, baselines changed continuously, and the development effort fell almost three years behind schedule. Incremental development will help refine system requirements, but the current increment must be baselined for this methodology to work.

System Complexity. Both of the incremental cases researched were very complex development programs. It appears that incremental development was an advantage for the development of complex systems. The increments allow the developers to break the systems down into simpler portions. SMDPS contains over 1.3 million lines of application code just to perform data preparation. Future blocks will incorporate interactive mission planning and mission material generation capabilities. All of these technologies are complex development efforts individually; together they comprise a very complex system. By dividing the system up, management is more likely to deliver the entire system.

While GWEN uses hardware already available, the complexity involves the software needed to manage a network of this size. The Air Force realizes that software is going to be a big investment, and has spent \$6.7 million for a software support facility. By concentrating on one capability at a time, the complex development can be conquered.

Subsystems. Both of the incrementally developed programs researched contained distinct subsystems that could be fielded individually. By delivering these subsystems incrementally the developers could focus their attention. GWEN was broken down into three phases that could be used

operationally. The initial connectivity capability connected Headquarters SAC with NORAD and a couple main operating bases. This core capability gave the user a chance to make sure that the development was proceeding in the right direction, while the developers validated the ground wave concept. The Thin Line Connectivity Capability then added terminals at the remaining bases throughout the continental United States. This core system is being used in the field today. Phase III will add more relay nodes, giving messages multiple paths to follow. The capability to select the best path is also being added during this phase.

SMDPS also could be broken down into smaller subsystems. The program initially delivered the capability to load data into the Air Launched Cruise Missile. Once that was accomplished, the developer was tasked with preparing mission data for the aircraft in SAC's inventory. Future blocks of the program will add the capability to automatically generate crew mission materials and, finally, an interactive mission planner to allow for mission planning at the unit level. These subsystems could be broken out separately to deliver a capability, and allow the developers to concentrate on one aspect at a time. The separate subsystems offered a good way for setting up the increments.

User Involvement. The user has been involved in both of the incremental cases researched. This is required for incremental development to help define lower level requirements. The user was involved during the initial operational test and evaluation of the GWEN concept validation and prototype systems. Write-ups during the tests and operational experience with the early increments has lead to the evolution of the Maintenance Notification Center, Software Support Facility, and other enhancements to the GWEN system.

SAC was actually developing portions of the SMDPS to maintain requirements flexibility. When the development effort proved to be too complex, they relinquished management responsibility, but retained their influence in the program. The user is getting hands on experience with the Phase II system, and has a process to submit requirements to Systems Command. This process is called the Material Improvement Program. Any requests submitted by the using command are reviewed and incorporated into later versions if deemed necessary. The users must be involved in this type of development in order to refine requirements for later increments.

Previous Systems. Neither of the incremental cases had previous systems from which to derive requirements, so increments were needed to perform this task. The Air Force never required data preparation prior to the development of smart weapons. Mission planning was always done manually. The user wanted to automate these functions, but there was no existing systems from which to derive requirements. They had a vision of what they wanted, but did not know exactly how to accomplish it.

The same is true for GWEN. The Air Force learned about the effects of an electromagnetic pulse on standard communications during atmospheric testing in the 1960s. They knew that the country's strategic forces had to be able to communicate after such an event, and started looking for solutions. One concept was to build a system that utilized a ground hugging wave. No such communications network existed in the Air Force, so a prototype had to be built to validate the concept. Since no previous systems was available, earlier increments could be used to understand what was needed.

Funding. Funding was a big concern in both of the programs developed incrementally. Incremental delivery allows the government to prove that the system works and to deliver limited capabilities within funding constraints. The GWEN program office decided to develop the system in

increments to employ the "fly before you buy" philosophy. The program manager said that, "This way you can validate that the system does what you want it to before dedicating a lot of resources to a concept that may not work. It would be cheaper to detect problems prior to spending money on an entire system."

SMDPS also took funding into account when management decided how to develop their system. The program manager feels that incremental delivery is the best way to get a realistic proposal from the contractor by allowing the contractor to level out manpower. Another benefit is that, if funding levels are cut, some capabilities can be delivered, and others can be shifted to later increments when funds are available. Incremental development allows management to deliver affordable portions of the program as soon as possible.

Development Expertise. In order to develop the complex systems requiring incremental development, software development experience is a necessity. Software experience was a big factor for SMDPS. The user wanted to manage the program themselves to maintain control. They had some experienced programmers, but had never managed a development program of this magnitude. Since SAC did not have development experience, the Phase III program was never baselined. Without a set of requirements to work towards,

the Phase III program did not progress to actual software design.

The GWEN project manager feels that the appropriate software acquisition expertise was available for its development. Unlike SMDPS, the requirements for each increment were identified prior to work on that portion. Development expertise lead to a more structured approach for the GWEN development program.

Summary. Neither of the incremental programs had a previous system from which to derive requirements. The developers broke the system down into smaller parts to facilitate development. Both of the programs were very complex; with multiple systems and an extensive software development. By working on one subsystem at a time, the contractors were able to concentrate on that area and to deliver an operational capability to the user. The program managers also indicated that the increments were done to work within funding constraints.

The differences were apparent when the researcher looked at development experience and requirements definition. GWEN management stated their system level requirements up front, and delivered the system incrementally to prove the concept and to field the entire capability. The SMDPS program had trouble defining requirements for the interactive mission planner. That,

according to the program manager, lead to an endless requirements definition cycle and a thirty-four month slip in the program. Development experience may have lead to the difference in requirements definition. GWEN had the appropriate people available to manage the development. The user tried to develop SMDPS in house. They had never managed an effort of this size, and had poor requirements discipline. The Air Force has learned from this mistake, and has restructured the program to break the program down further, to identify requirements up front, and to give responsibility to a development organization.

It seems that incremental development is appropriate when:

1. There is no previous system
2. System requirements are defined up front
3. The development is complex (extensive software)
4. The user is involved
5. Funding is not available for entire system
6. Subsystems can be fielded separately
7. Development experience is available

Conventional Methodology

The same type of analysis can be conducted for conventional programs by comparing the two conventional cases; DFMR and REACT. The impact of each of the characteristics are described below.

Requirements Definition. One of the assumptions of the waterfall model is that the system level requirements are known at the beginning of the program, yet only one of the conventional cases researched had a thorough understanding of the system requirements from the beginning. DFMR derived its requirements from two systems already in the field. The requirements are to integrate technology from the B-1 receiver and the airborne GWEN system, and to make the system work for the Peacekeeper/Rail Garrison.

REACT had problems with the definition of their requirements; and this has led to technical problems and a cost overrun. The current launch control centers rely on a lot of manual processing. The plan is to automate all of these processes concurrently, because SAC's number one priority is fielding a system as soon as possible. The project manager indicated that they did not have a systems analysis done prior to contract award. Requirements are being redefined as the system proceeds through development. Time should have been taken prior to contract award to

identify the system requirements. That is the only way that a top down development will work.

System Complexity. The conventional methodology did not seem appropriate for complex systems requiring an extensive development effort. REACT is turning out to be more complicated than expected. Just on the command and control portion of the system, the contractor must integrate higher authority communications, rapid message processing, and error correction. All these functions must be implemented using software. The project manager estimates that 70% of the effort is concentrated on software. The code has grown during development and has created a memory reserve problem. The project manager believes that maybe the contractor did not have a strong understanding of the requirements in the beginning. This portion of the system also has to be integrated into the Weapon System Control Element, which is evolving concurrently.

The DFMR development program is less complex, making it perfect for conventional development. The major portion of the effort deals with the integration of existing capabilities. The software consists of only about 10 thousand new lines of code; the remainder of the software was developed for MRT. This allowed the Air Force and the contractor to plan the program early, and stick to that schedule. Based upon these cases, it appears that a program

should only be developed conventionally if the system is relatively simple.

Subsystems. Cases with multiple subsystems appear to be too involved for the conventional methodology. The REACT program has multiple subsystems. First of all, the program can be divided into two separate development efforts; WSCE and HAC/RMPE. Both of these efforts also consist of some distinct capabilities. HAC/RMPE is working on higher authority communications, alarm integration, rapid message processing, and error correction. The WSCE portion is developing NCA message processing, rapid retargeting, and console development. In essence, several subsystems are being developed concurrently in the hopes of delivering the entire systems earlier.

DFMR only consists of two boxes. The receiver was already developed for the MRT program, but must be retuned to new frequencies. The second box must be developed to incorporate capabilities from the existing airborne GWEN system. The development effort could not be broken down any further, so the conventional development method was employed. Unlike REACT, DFMR had a limited number of subsystems to develop, so conventional development was more appropriate.

User Involvement. Since the requirements are supposed to be defined at the beginning of a conventionally developed program, user involvement can be limited to ensuring that requirements are being met. The user had to be involved in the REACT program from within the program office. Personnel with prior experience as launch control officers are employed as project managers to help ensure that the user's needs are determined. They are in contact with their peers on a daily basis. This has been important because of the lack of requirements definition at the start. With this environment, the project managers can work with the users to determine the actual requirements as problems arise.

User involvement has not been as important in the DFMR program. The user helped to define the system requirements at the start of the program, and the development effort is fairly straight-forward. Even with the lesser user involvement, DFMR has remained on schedule. The program has been successful because the user and developer took the time to identify needs at the start of the program.

Previous Systems. If a previous system is available, the program is more likely to be successful with conventional development. Only the DFMR program had the benefit of working with a system already in the field. The program is building on the receiver currently flying in the B-1 aircraft. The GWEN capabilities that are being added

are also in production, making the integration of these technologies easier. There have been some problems with technical data from the previous systems, but the equipment is available.

The REACT program is making modifications to the existing launch control centers, but the additions are something that has never been done before. The changes are to automate some of the processes now being done manually. As the project manager said, the Air Force has upgraded the missiles in the silos, but has never worried about the launch control centers themselves. The HAC/RMPE portion also has to be integrated into a console that does not exist at this time. Problems have occurred because there was no previous system to derive requirements from.

Funding. Funding was not an issue in the conventional programs. While an incremental program like GWEN cost the Air Force half a billion dollars, REACT is only a \$36 million program. The government could afford the entire REACT and DFMR programs in one increment.

Development Experience. Even though good conventional programs are not complex and requirements are known, the program office should be manned with experienced personnel. The REACT program office had some problems with software experience. They had a software manager with six years of

experience, but lost him to another project. He was replaced by somebody with very little acquisition experience. Acquisition experience is important in this program, especially when software makes up 70% of the effort, and requirements are being defined after contract award.

There is more development experience on the DFMR program. The contractor has done work on the receivers for the SAC airborne command post and the Minuteman launch control facilities. The program manager feels that the three technical support contractors and six government employees are sufficient to manage the program. That has been true so far, indicating that experience is required for conventional development.

Summary. DFMR had good results using this methodology. The purpose of this program is to deliver a system integrating technology from two previous systems. The requirements were known at the beginning of the program. It is a fairly straight-forward concept that was comprised of only two subsystems. Funding and manpower were also sufficient for the development of the entire system.

REACT has had technical and cost problems employing the same methodology. The modification is very complex; incorporating many new capabilities concurrently. The user has been involved from the start, but the effort went on

contract without taking the time to define requirements. The situation was made even more difficult when the program office lost its software acquisition experience. Despite these issues, since the funds were available for the entire program, the effort was planned for concurrent development and production.

The factors for successful conventional development appear to include:

1. A similar system already developed
2. Requirements definition at the beginning
3. A straight forward development
4. Experienced personnel
5. A limited number of subsystems
6. Funding available for entire program

VI. Recommendations

The conclusions from this research should be viewed as recommendations until confirmed by future research. This research suggests some patterns concerning the development of command and control programs. The purpose of this research is to identify program characteristics that may help to determine the better development methodology. While there is literature available proposing such a list of characteristics, little research has been performed. This research used the case study method to explore actual command and control programs, in the hopes of identifying key factors to make this determination.

Implications for Development Managers

The lists provided in Table 2 below could be used by development managers to help determine the appropriate development methodology for future command and control programs. After categorizing the program for each of the characteristics, the manager can compare the results to both of the lists. Only if the program meets all of the characteristics should that methodology be used. Any differences should be rectified prior to awarding a contract for the effort.

<u>Conventional Development</u>	<u>Incremental Development</u>
Previous system fielded	No previous system
Defined requirements	System requirements known
Simple development	Complex development
Experienced personnel	Experienced personnel
Few subsystems	Usable subsystems
Funding is available	Can not fund entire system

Table 2 - Selection Criteria

Such a list might have been helpful to the cases that ran into development difficulties. If this type of list was available earlier, the SMDPS managers would have realized that they needed to acquire personnel with software acquisition experience, and to break down the system further to define requirements. Experienced management have now decided to utilize additional increments to facilitate system development.

Using the same list, REACT management may have appreciated the complexity of their system. That appreciation would have led them to decide against convention development. The system design appears to be too complex for simultaneous development of all capabilities. The user would benefit more by delivering a core capability initially, and then building on that capability. This research also shows that, even with incremental development,

management should would need to take the time to derive requirements and to obtain the appropriate personnel prior to contract award.

Implications for Researchers

This section describes the implications that this research has on research methodology, and future research. Many lessons were learned during this research.

Some disadvantages of the case study research methodology were outlined in chapter 3. After conducting this research, it seems that many of these disadvantages can be avoided by making the study systematic through preparation, conduct, and reporting. Prior to data collection, a literature review was done. The review lead to some ideas about what characteristics were important when determining which development methodology to employ. Copies of program historical reports, program manager's assessment reports, and contracts were obtained to learn more about the development efforts. The assessment reports were especially helpful since they were very honest in their interpretations of the program's progress. Using the information available, an interview guide was created to be used for all four cases. Once the cases were categorized by methodology and progress, a comparison of characteristics was performed. The resulting pattern became the basis for the recommendations made by this research.

Researcher bias is sometimes seen as a shortcoming of the case study. This was overcome by using the same interview guide for all of the cases. In this study, after the literature review, the researcher developed a preliminary list of characteristics that were thought to be important to the selection of development methodology. Bias was limited by allowing project managers to speak about how those factors impacted their program, and to bring forth any additional characteristics. The informal nature of the interviews allowed the person being interviewed to do most of the talking. The interviewer only spoke to clarify statements or to ensure that all data was collected.

Sometimes the validity of case studies is questioned, because of the possibility of confounding variables or case alteration. While this research was not trying to show causal relationships, the explanation building approach used helped to confirm the original expectations of the researcher. The researcher assumed that other variables were constant, because of the similarities between cases. The cases were selected from programs managed by the same organization, within the same time period. The conduct of the interviews should not have altered the settings, since all interviews were basically identical. If the cases were impacted, the impact was assumed to be equal.

Future Research

This research should be considered a starting point for future research efforts. Further studies should be conducted to find additional characteristics that impact the determination of the appropriate development methodology. Only those characteristics originally anticipated by the researcher were taken into account, since no new characteristics were pointed out.

This research was limited to command and control programs. While this method may apply to other programs, additional technologies could be taken into account when conducting future research. Including other technologies would help to strengthen the external validity of such a study by ensuring that the same characteristics are important. In this way the findings from the research could be generalized across a larger set of programs.

As new technologies arise, neither of these development methodologies may be appropriate. New technology is defined as technology that has not been used operationally. Incremental and conventional methodologies both require that the system level requirements be defined at the beginning of the program. With technology that has never been used before, the user may only have a "vision" of what they need. Development methodologies like Rapid Prototyping may be more appropriate in such a case. All of these factors

should be researched to develop methods that are usable for all programs and development methodologies.

Appendix A - Interview Guide

1. What project do you currently work on?
2. How would you describe the system?
3. What is your job title, and what responsibilities do you have?
4. Is the project being developed traditionally or incrementally, and why?
5. What approximate percent of the contract manhours does software development represent?
6. How many Lines of Code are required?
7. Has the contractor been meeting cost/schedule baselines?
8. Does the product meet all specification requirements?
9. How many other systems does the project interface with?
10. Can the system be broken up into stand-alone parts?
11. Did the user/developer understand the system requirements/design up front?
12. How was the user involved in the systems development?
13. Was there a previous system already in the field, or is this a totally new concept?
14. How many software development personnel did the project have assigned to it? Did they have any previous experience?
15. Did funding levels impact how the system was developed? (Example: not enough money to fund entire project?)
16. Have there been any major changes to the program direction/requirements? How were these changes determined, and how were they implemented?
17. Was there a good understanding of the tasks required prior to contract award?
18. If incremental, did the contractor perform better with each increment?
19. If incremental, what were the technical and political benefits of the increments?

20. What other factors influenced the decision for development methodology?

21. How do you feel about the method used by your project?

22. How do you think the project would have done if an alternate methodology was used?

Appendix B - Transcribed Interviews

DFMR Program Manager

**** Is the DFMR program being developed Traditionally or Incrementally? ****

A little bit of traditional, a little bit of incremental. The traditional being that I'm essentially going to build one receiver, and in FSD I'll have eight receivers at the end, as close to the production version as possible. The way my SOW is written in FSD he's essentially developing the engineering drawings that he will use in production. He'll use these eight receivers to incrementally achieve the final system. System 1, he can make some improvements and see those in receiver 2. It's a pretty fast waterfall, so he doesn't have a lot of time between receivers. In fact he is already building receiver 2 before he finishes receiver 1. If you looked at the schedule it would be a traditional waterfall schedule. However, we are at the tail end of another program that is MRT, which was built for the B-1. The B-1 receiver was designed for an aircraft application to fit on another airframe. When the Peacekeeper folks said we have to have the same capability on a mobile train ESD said that we can build you a whole new receiver to meet your requirements and your cost will be X, or we can take the MRT receiver and make some adjustments to that receiver. Build on lessons we

learned when we built it. In a faster time and for less money, we can give you a receiver for your Peacekeeper Rail Garrison requirements.

** You said that there are eight receivers being delivered, what are those? **

Yes, eight units will be delivered in FSD, and then we'll have receivers for production. They only needed forty some receivers for the peacekeeper Rail Garrison mission. SAC also need receivers for the Minuteman Launch Control Centers. After we were funded for Peacekeeper Rail Garrison, SAC said maybe we can use that to replace some systems that we have that are getting older. Incrementally, we're making some changes to the design of the peacekeeper application so that it will also work in a Minuteman Launch Control Center, namely in the form of an antenna modification. So. I'm a little bit of both. If you look at the bigger picture I'm incremental. If you look at just the program DFMR, you would say traditional.

** Are the eight FSD receivers going to go operational? **

We would expect as we look at the schedule today to make them go operational. There is an option in the contract to allow us to send them back to Westinghouse at the end of FSD and have them brought up to a production level. Any changes that we made would be put in there. The way that they bid it was that there would be nothing more

than to paint them and touch them up, however. They did not put a lot of money in the budget for ECPs.

** What is going to be done with the FSD units, is the user going to be able to play with them? **

He's going to get to play with them for IOT&E. Then we'll send them back to get refurbished.

** Do you know how much software is being written for the program? **

Very little. Again, it was written under the MRT program. Total new code is only 10K lines of code roughly, versus 40K total.

** It seems like that's a pretty small part of the program. Is that because this is a communication system? **

Right. This isn't a real complex system, but it does everything within its box. It's a 186 processor inside it for one half, and an 8088 in the other half. The receiver in the B-1 received on one frequency band, but has excess capacity to possibly receive on a band that was real close to it. the original plan was to tweak the reception part, and instead of tuning to the original frequencies, they would tune to different frequencies in another bandwidth for what's called GWEN.

They had developed an airborne application. When we went on contract we did not tell vendors you have to use the GWEN airborne system. We just said we want you to take the capability of GWEN and put it into this receiver. Rockwell

International, who built the B-1 receiver, came up with this whole scheme of things anyway, and said we've been playing in the lab and if you ever wanted to do a ground platform and not an airborne platform we'd have excess capacity if you take a couple receiver cards and tune them to another frequency. There's enough capacity within an executive processor to handle more functionality, so we put a couple other parts in and take some parts out and you'd have both the low frequency band plus you'd have a very low frequency band. You would be able to receive in both networks without changing your box. When reality came around it was a little different, but not much. Vendors said yes, you can do that. We can reuse all the code that was developed in the MRT system. The eventual winner was Westinghouse and they teamed with the original contractors, and their reusing all the code so there's very little new code being developed for this receiver.

**** How is the contractor doing cost wise? ****

If you looked at the cost reporting, you'd say he's losing his shirt. It's not because it's harder than anticipated, it's because we put him into a must-win situation. Westinghouse has always been a competitor with Rockwell in this area, and when the MRT system was being competed for the B-1, Rockwell and Westinghouse were in a fly-off. Westinghouse had originally won the contract for the receiver that is currently installed in the SAC Airborne

Command Post, its installed in all the Minuteman launch control facilities. they thought they were going to win the B-1 contract because they had all this previous knowledge. Rockwell was trying to come in and enter the market, and under bid them with innovative pricing maneuvers. They lost their shirt on the B-1. It's a fixed price type contract, and they are losing money today as they produce. They are not making the profit they thought they would. Westinghouse did not want that to happen. They lost on the B-1, and didn't want to lose again. So every trick Rockwell tried on them and won, they thought Rockwell would try again. They learned their lesson and did every one of those, which put them into the running for this, otherwise Rockwell clearly had an advantage. They probably went too far, but they didn't want to lose. He thinks they still have a chance of breaking even on this. They're still going to overrun, but he doesn't think they will make as much a profit. When it gets to production they get some of that back. They're losing, and part of that is because the government is a little overzealous as we enforce the rules.

**** Do you mean going to the lowest bidder? ****

Well not just taking the lowest bidder. Once we go on contract, we want more than we originally went on contract for. things change, so we see what we can get the contractor for and he's trying to fight that with his staff and it's a never-ending battle.

**** Is the receiver meeting all specification requirements at this point? ****

We haven't tested any yet, but they've gotten to the point of building what they call receiver #0. It's not receiver one of eight, it's before that. they've done the first machining of the chassis for the MRT side, and the GE folks have done their first casting for their receiver. The cards were essentially already built, their just changing a little of the technology that was used in going from type to another. Component placements are the same, it's just different manufacturing technology. They've gotten it up, and it's at least processing messages. The software isn't fully function yet, but their achieving success. They still have to marry a few pieces together yet, so there is still some integration problems.

**** Are they on schedule? ****

Their not on schedule, but it's funny because we asked them to do a 36 month program in 30 months. they'll probably come in a t 36 months, and part of that is because contractors take a little longer man up than they think it is going to take. They lost two to three months just getting people on board, where the schedule assumed they'd have people immediately. There were some issues to deal with as far as what we specified. We fought some of those battles along the way. They realized that they underbid, but they wanted to hold the number of people they bid, they

didn't put enough people on at the right time, did not peak when they should have, and they fell behind and had to catch up. They're three to six months behind.

**** Were the requirements known up front? ****

In a general sense - yes. In a specific sense the answer is no, because even though the GWEN system was going into production, as was the MRT system, even on the GWEN side the Government documentation was weak. We don't fully understand what's there. So, all those things in total, I think the GE side thought they were doing a little bit different than what the Government thought. And Westinghouse was somewhere in the middle on the MRT side. We thought there was going to be documentation there that wasn't there. From a requirements standpoint, it wasn't that. When Westinghouse didn't know what was going to be there, they've had to fulfill that requirement. We were timely in getting it to them, which didn't impact them, but they'll say it did. We over-specified in some areas and under-specified in others. When we tried to correct some of those we made mistakes. But in a general standpoint, all requirements were met by both parties.

**** Has there been any engineering changes? ****

We're coming up on two years, and we're only up to ECP 11. So, no, there hasn't been a whole host of engineering change proposals. I don't know if that's a lot or not because it's my first program. I've never worked in FSD.

I've worked production programs before, and they didn't have a lot of ECPs. I don't think we have a lot of ECPs, but we don't have control of the design yet, it's still in Westinghouse's control, nothing's been established. Once we baseline it, we could see that number mushroom very quickly. the system requirements have changed eleven times.

** Do you think doing it traditionally was appropriate? **

As I was saying, this is kind of a hybrid. If you look at platforms I'm traditional. If you look at a receiver and what it will do I'm not traditional.

** The new receiver was traditional, and I don't see any way that they could have broken it up into increments. **

There's only two boxes. The first box was developed under another program. So, as I get to the second box, I make some changes to make the second box fit with the first box. The problem we've dealt with is that nobody is happy with the 186 processor. One of the reasons that Rockwell did not win is that they listened to the user and proposed a 386 processor and a lot of other bells and whistles. They lost focus as to what the RFP actually asked for. They were all valid wants from the community. It's hard to say a 386 wasn't a better idea than the 186. But as soon as I made that jump I could no longer be an incremental, I became a traditional. As soon as I changed processors I could use the same software anymore. As soon as I change the crypto device, that's half the machine, so is that a little change

or a big change? It's not like on an airplane where I've got an F-15 and I go from one type of an engine to another; or even if I'm going from one model engine to another model engine. Am I the same airplane anymore? Sure, because I can plug and chug different engines. But I look at the box and say it still looks the same. So, yes, that's your incremental. I think incremental has its advantages as long as you understand what it is you are doing. It becomes very difficult because you have this large community out there saying I want my little increment out there first.

****Do you think you have sufficient people with software experience to help develop this? ****

Unlike ASD where you have this big conglomerate of military and civilian engineers, we at ESD don't have that big conglomerate of people. I use what's called a Technical Engineering management group. It's not MITRE, we use another company called ASET. So, yes, I have three software engineers that work for them. That's more than sufficient for the amount of code I have. I think the seventeen man years of support contractor effort and essentially six military/civilian slots I have are sufficient.

GWEN Program Manager

GWEN is a Ground Wave Emergency Network. It uses a ground-hugging wave, it doesn't use the ionosphere to bounce then wave back down to the earth. In an EMP scenario, a nuke could be detonated over the US that would charge particles in the ionosphere, and standard communications as we know today would not function because that fraction of a second that huge blanket of energy would come down on the earth and anything that is not shielded from EMP would be dead electronically. Things like your radio, tv, car - anything that can conduct electricity; high power lines would tie onto that and just burn out the electrical components. GWEN is designed to be immune to those effects, because it uses a ground-hugging wave, it doesn't use the ionosphere.

It is a strategic system. It is the only system that ties the National Command Authority, the president, into his strategic sensor sites. In turn NORAD can review the data from other sites, provide that information to the President. The President can decide what type of appropriate response is warranted, and then relay those instructions out to the strategic forces. We're talking about land-based ICBMs, tankers and bombers.

** What was your job on that program? >

I started out as the Thin Line Conductivity Capability and the Relay Node Expansion program manager. From there we became the Deputy program manager for GWEN, which

encompasses an airborne capability, where SAC's EC-135 Lookingglass has the capability of injecting messages into our relay nodes.

** The system description mentions three phases; what are those? >

There are three phases to GWEN. First, we had the initial conductivity capability. The concept of fly before you buy; let's take a look at the technology and demonstrate it before we commit large amounts of money to dump into a system that may not be what we want. That was done. It tied in some command post with some of SAC's main operating bases. So at the command post you had an input/output station where you can inject messages into the system, and also receive messages out of it. At the other end of the spectrum you had the SAC main operating bases where all they had a receive-only station, where they can receive information off of the network. Between those two points you had a series of relay nodes, kind of like a relay race where you're passing the baton from a person to another person. Because you use a ground hugging wave, the distance between the relay nodes is only about 200-250 miles depending upon the ground conductivity. In as such it will dictate how far apart you can station your relay nodes. That initial phase of the program was demonstrated back in the late 70s, early 80s.

From there we got the green light to expand the system, to go into FSD. We entered into the second phase, which is called the Thin-Line Conductivity Capability. Basically what that is it made a figure eight around the continental US where in the center is where Headquarters SAC is located.

the third and final phase of the program is the Relay Node Network Expansion, which adds forty additional relay nodes into the network. You can think of it kind of like a Ma Bell telephone network, where you want to call your mother in Florida; here you are up in Boston. The most direct route is just going south. Because the phone lines may be tied up, your telephone call may be rerouted up to Oregon, down to San Fransisco, over to Colorado, back down to Texas, and then over to Florida. What we're doing is adding forty production relay nodes into the network to give additional disjoint paths that these messages can go across the country.

** Why was that done at a later stage? Was phase three done due to funding considerations? >

What was originally scheduled to be a much larger network, it was envisioned that we would have about 250 relay nodes across the country. Because of geo-political issues, GWEN by far is the most geo-politically sensitive program in the DOD. We had concerns from Congressmen who were receiving letters from their constituents saying why do we need to do this? You also had concerns with some

journalists writing articles about high-power tension wires and the EM fields that the high power tension wires transmit and emit, and how that may or may not be linked to leukemia. So, there were problems like that, and as a result we had options built into the contract. We exercised the options on time, and it was just the program was laid out. We had delays because of geo-political concerns on the country.

** How much software was involve in the program? >

I don't have the total number of lines of code, but we had a significant, that's a relative term, amount of software. We had both hardware and software. We have a \$6.7 million software support facility that we're building just to support the system software on this program. The program is valued at approximately half a billion dollars. There's a sizable chunk of money. There's a Lt Mark Hughes who is the software project manager who could help you.

** Did the contractor have any problem meeting cost and schedule baselines? >

It was a Firm Fixed Price contract. He did not have any problem meeting his program milestones. On the contrary, it was the Government that had problems delivering the GFP, in this case we're talking about land. In Massachusetts I'm about three years behind schedule. I have two relay nodes that still need to be installed. Two congressmen have made it very difficult for us to deliver the product, taking consideration, and rightly so, the

constraints we have with the National Environmental Policy Act of 1969, and what we must do to preserve the environment. The Air Force is 100 percent behind that. But every time we move forward, they attempt to push us backwards.

** Has there ever been a system like this before? >

No, GWEN is the first of this kind.

** It seems like the Air Force knew what the requirements, is that correct?>

The Air Force knew what the requirements were. The phenomenon of EMP was demonstrated back in the early 60s when we did atmospheric testing over Johnson Island, we had problems with electrical components burning out.

** Was the initial phase conducted to make sure that the system fit all the requirements?>

Yes, it was to demonstrate that the concept did indeed work. You fly before you buy. There is a lot of things to say about that. On the other hand, we can go back to my days when I was working on space communications systems. There, when you're buying three or four satellites, it's very tough to build a full-up system where you can demonstrate that capability. You can do some prototype designs, and you can take certain subsystems and play with them. In order to take the system and do a design on it, you have to spend some up-front money. When you're buying

one or two it doesn't work. If you're buying a larger quantity of items it makes sense.

**** Was the user involved in the initial testing?>**

Yes.

**** Has there been any major changes to the program since it started?>**

There have been some changes where the GWEN Maintenance Notification Center has evolved into more than it was originally intended. It does more than just monitor the status; it allows you to do a query of the system, verify that your neighbor relay node is up, if there's a failed neighbor report there are sensors in it; where if somebody tries to break into the system it will protect itself and hibernate and the classified information inside the system will blank out so there's no compromise.

**** Was the whole program funded up front, or did they do phase one first and then go back for money for phase two...?>**

Phase 1 and Phase 2 were both using R&D funds. With R&D funds you incrementally fund the acquisition. With the third and final phase, the production effort, we had production dollars and we funded that effort up front.

**** Where they separate contracts?>**

The second phase and the third phase was one contract for the hardware. Because of the SECA competition in contracting act we had to compete the physical installation

of the forty production nodes. As a result of that GE, the prime contractor, lost out to Contel. Contel will be the installation contractor.

**** What did you learn from the first phase of the program?>**

You're going back to the early 80s, which is well before my time, and I really can't talk about that. Obviously there had to be things learned from that. If the technology was not there or cost too much the Air Force would have to make the decision to press on or to cancel it because of cost.

**** How did incremental development help the program politically?>**

The mindset is that the Berlin Wall is down, and the Soviets are disarming their tactical forces. There's not a threat anymore. There's a lot less going on now in East Germany. The fact of the matter is that the Soviets have dismantled part of their tactical forces, and it is the tactical force where you spend the bulk of your money. The strategic forces continue to build up. GWEN is a strategic system. It is designed to provide the NCA the ability to communicate with strategic forces. By being able to do that you act as a deterrent because you can now talk, and the threat of a high-altitude nuclear detonation cannot deter your ability to respond.

**** Do you think that structuring the program into three phases was appropriate?>**

It is the right way to go. There is some discussion as to whether it would be more prudent to give the entire program to a civilian contractor, and have that contractor install the system as opposed to having the Air Force getting involved in doing the work. The civilian contractor does not have to comply with federal regs, environmental regs - the restrictions placed on the Government. Example; right down at Westport Mass. there is a 300 foot tower that was installed this time last year. They did not have to do a visual historic assessment. Around that part of the country you've got a lot of architecture that has certain historic value. the public was behind that, but they weren't behind the DOD because of all the high taxes you pay. When the DOD comes into their backyard to do this type of work the vent their hostilities.

REACT Project Manager/User

REACT came about with the thought of being able to target a lot of the Soviet mobile ICBMs. We needed to be able to keep up with real-time targeting of these mobile systems. The system we currently have out there with the Minuteman IIs, and the Minuteman IIIs, we didn't have the capability to do that. It was antiquated technology, reliability was going down, maintainability was becoming difficult. So, we needed a new system. We've always updated the business end. We've had three generations of missiles, but we've never really done anything to the launch Control Centers or added any creature comforts. We just kept plugging comm systems into an enclosed area and the person always had to adapt. REACT took a long, hard look at how to work all these things in, and provide some creature comforts.

With this program specifically, we went from the time the RFP was out to the time the contract was awarded in about 117 days. So, it was pretty quick to go through source selection, and at that time it was the quickest turnaround for a source selection in Systems Command. Because of the timing, we've done a lot of things in parallel. Where normally you would think you would finish your engineering phase, and then do some testing, and then make a production decision after you DT&E. Because of the timing of this we're making a production decision before DT&E. If you hit and you're on the mark, you're ahead of

the ball game. But, if you make a production decision and find out during DT&E that you have a lot of problems, then it puts you back behind the power curve. For whatever reasons, the management decided to make this program parallel. We were at risk, and you find out a lot of times that because of the engineering phase that we're in now and no testing, you have to rely a lot on analysis that the stuff is going to work that they're putting together. I think that is the biggest risk, because we do some incremental testing, but not really the system before we make a decision. Understand that both people are making a production decisions separately. Loral and BMO are making their production decision, and we're making ours too. With this dual product division, we can make a production decision and find that DT&E is working well, but Loral, when they make their production decision, if they have problems, we've got equipment that's on the bench that's not being incorporated. So, that's tough for us.

** You don't know why they decided to do development and production in parallel?>

I personally don't know why they did that concurrent engineering. I think it was because of the threat. We needed this thing fielded ASAP, zero delay.

** Was this supposed to be a relatively simple system; maybe that would lessen the risk?>

The requirements weren't really known. Usually what happens, the way I understand it, is that a lot of your operational requirements analysis should be done before you go out and do the actual engineering phase. We didn't have that luxury, because we did most of the requirements definition during the time when you should be driving out your lower level specifications. We were really doing the top level detail at the same time. Usually you should have the top level specs done and then you drive out, during your operational requirements analysis, your lower level requirements. We we're doing all that in serial. We didn't have the luxury of a well-thought out requirements done for us. So, we had to do that on our own with SAC. SAC was the one that was driving out requirements.

** Is there a lot of software for this program?>

The program specifically is very software intensive. Not so much for Loral, because Loral is more hardware and rehosting the software that is already out there. We had to develop almost totally new code, except for some algorithms that were there from a previous message processing program which really didn't come to fruition. So, we've learned a lot from (that program), and we tried to preclude some of those mistakes again here. The HAC/RMPE portion of this product is very software intensive. Software is really probably the highest risk area in any program. We're trying to be smart about it, and not put ourselves in a position

where , when it comes down to FQT we haven't done some incremental testing, that it doesn't give us a high degree of confidence when we go into FQT that it's not going to work. We're trying to do some smart things in the software arena right now, in different builds of software, different releases.

**** Do you know how many lines of code are involved?>**

Comparatively speaking, it wouldn't be as large as your program, the SRAM II. I'd say 70% of the program is software. You had to go out and get processors to run the stuff. It's more or less 60-70,000 SLOC. It's the majority of this program. It's not a lot of lines of code, but it's very intricate because of all the message processing and error correction. There are rules we have to follow. It has to be very precise software and most of the effort is software intensive by far.

**** Do you have software people on board in the SPO?>**

That's a good point. Most of our software expertise comes from our MITRE support. Because this is supposed to be done in Ada, there's not a lot of Ada experience in the blue-suit community. We brought one individual on right out of college who's supposed to be our software guy. we had one Captain who was snagged by Mission Planning to do the B-2 stuff. He was really our software guy; he was an AFIT graduate and Academy graduate. He had some understanding, he had six years of acquisition software. He was really the

right guy for the job, but then they snagged him from us. Really, we don't have the right people from the blue-suit community to do the software for this program. So, we have to rely on MITRE.

**** How has the contractor been doing from a cost and schedule point of view?>**

Like I said earlier, GTE is in a loss position. When you bid competitively, you bid on a couple of things that are supposed to fall in your favor. If they go your way, you're golden. But, if they don't, you find that you put yourself in a loss position. right now GTE is not in the best financial position for us to deal with them. One of the things that hurt this program was that this Engineering/manufacturing phase was a fixed price, incentive fee contract. I don't see how the Government ever let this go out as a fixed price incentive fee contract for an unknown entity. When you develop something, it's development, you can't ask these guys to have a crystal ball and look down the road and know exactly how much it's going to cost. Especially if problems crop up. Especially when people compete in a competitive area, you try to take an optimistic look to win the contract, that's business. GTE did that, some things popped up that we didn't know about. Some nuclear testing showed that some of the things that GTE had planned on was not as survivable. But, because it was

fixed price, it was their nickel. So, from the very beginning of the program, they were behind the eight ball.

** How about technically, I read about a memory reserve problem>

Personally I think the program is a) going to get fielded, and b) do what the Government wants it to do. Not that there's not going to be heartburn along the way, because there will be. There are some things that the Government has had to go back and relook. It comes time to a technical point, do you cancel a program because you want gold-plated faucets? Will a stainless steel faucet give you what you want? It's easily perceived that you can get everything you want at contract award because everybody is so hyped up that yes, we can give you anything you want government, and the Government says, well, we'll take anything you can give us. The reality of it is that as you go along and as you define your requirements, you realize that maybe what you had thought because we didn't have those well-defined requirements in the beginning, you find out that to get something like this it costs you schedule and it costs you money. We don't have the luxury of either one of those. So, you have to take a step back and say, ok, what is the reality of it? Do we potentially lose the program because you wanted some gee-whiz kind of gadget, or do you take a realistic look at it and say I can do the same mission with something that is not as fancy, doesn't have as

much flair? That's what we've done right now. We've taken a look back at the A Spec and said, ok, we're out here pulling alerts, does this really provide us anything more? I think if you had pure engineers without that ICBM background they would say that's good engineering practice, but they don't understand the impact of whether or not we can live with it. If we can live without it, and SAC sometimes said, ok, I need this because maybe it was something they didn't think of, in turn said we don't really need this either. There's been a lot of horse-trading going back and forth.

There are some show-stoppers. We almost had one with what's called Nuclear Weapons Effect Recovery. There was a requirement that you couldn't lose any data through any natural or induced nuclear environments. The truth of the matter is nobody really understood a lot of the testing that was done at White Sands - what this processor was going to be able to do. They did some preliminary testing and this thing was getting shot to hell. So, we go to SAC and say your requirement is don't lose data, and it has to operate through it. For you to do that, you have to get space-tested parts that will cost you about \$70-170 million and a three year slip in the program. What do you want? You work around and change the spec. That's the requirement we're talking about that we relaxed it, but was it a realistic requirement to begin with, with the time and money the

Government was willing to invest? Technical issues have come about. We've been able to dance with them a little bit and make them work. I don't think there's any show stoppers right now. This memory problem can be, but there's options for us out there. maybe like a high-density card. We have an expansion slot because we have to provide for growth. We don't want a system that's not robust enough to meet future comm systems. Those kind of things have happened, and we've been able to work around them.

** Why is there a memory problem, is the software effort bigger than they thought?>

Yes, we didn't understand the complexity of the software. We didn't understand a lot of the requirements that were levied. Especially in the human factor arena. Because we live with this antiquated system SAC said, ok, these guys have been pulling alerts 24 hours a day. Lets give them something that is user friendly... All that is software, and that's a lot of code. Maybe GTE wasn't smart enough. When you compile it, it grows. Maybe they didn't have a good enough model because they didn't have a strong understanding of all the requirements in the beginning. There's fault on both ends. They didn't guess right is the bottom line. They didn't guess how big the code is. The code grew. As the code grows, you find out that it potentially gives us the growth problem. We've got to work around that. You can look at somebody and say, you go make

this thing work. Then what? Then he's got to defend why he needs some more money at \$50 thousand a card in front of some Congressional panel. They're going to go, what, you...gone! The management's in a position where they have to orchestrate smart versus being practical. We find that because we don't have all this backing, we're at the mercy at any give time if we show potential show stoppers, they'll cancel us. They don't have any qualms of just zeroing out our program and buying another B-2 tire.

** Do you think the program could have been broken up?>

Yes, first of all there should have been someone to take the lead. There should have been an integrating SPO. I think that would have solved 90% of the problems. Do something that says, when it comes down to making a decision, somebody makes the decision and both people march off sharply. Not marching off in different directions, because six months down the road, now you got a bigger problem because you didn't avoid it up front. Now you have contractual issues on both sides, and the Government ends up paying. We didn't do that. We let these guys dictate a lot to the Government. Because we're relatively new to the acquisition world and don't understand all the things you can't do, how that can't be changed to make things happen. From an operational background, I've got a DO community. If you've people fighting amongst each other the DO says you shut up, you do this, and you go do this...and we salute

sharply and we step off. We don't have that kind of decision-making capability in this program.

** How was the user involved in the program?>

We are a user. We have to be able to put on a dual hat. Everybody wants all the lights, and bells, and whistles that you can possibly get. If I could buy an Alfa Romeo for \$12,000, I'd be an idiot not to ask for an Alfa Romeo. Somebody has to say, time out, you're getting really a Jetta. Will it get you from A to B? If that's the requirement, it doesn't mean you have to go from A to B with a CD at 120 Watts. That's a problem we're finding early, because there was no well-defined requirement. We don't want to become another system that never got fielded after spending millions of dollars on it. We find it very difficult to look at our brethren and say you can't have it. It would be nice, but we don't have the money.

The user in this program is intimate with the program. We call on a daily basis, because we're such a tight community. It's good from a point of view that we can go to them and tell them that the reality is that they can't have it. I know you're going to have to fight for it because that's your job. My job is to is to tell you that you can't have it. We have a tremendous working relationship with the guys at SAC; that's where we came from. I think that's the best thing that ever happened to this program.

** Have there been any major changes to this program?>

There has been two major changes so far. One - there was a communications drawer called SACDIN; we were supposed to go in that rack. We've changed that so now we go into the WSCE console. The other thing was with this nuclear weapons effect. We had to orchestrate a lot of different design changes. We had different chips that had to be put on board. We went from a single disc drive to a dual floppy. We had to do some additional testing. The recovery procedures, what were the requirements to be able to save X amount of messages, and save these types, and save them to a disk, and be able to recover them automatically. I think those two were the two biggest ECPs. Because we're only \$36 million, this was a tenth of it. We haven't had any major changes where you'd have to redirect and scrap everything you've done, because we've been able to catch them early.

** Have they impacted the schedule?>

The funny thing about it is that all the perturbations have not changed the IOC, yet. There's been internal slips, but IOC has remained the same. The question for us is, are we going to be fielded with this AEM from Rockwell?, because their a year behind us. Are they going to be able to provide us with this system so that we field concurrently and not do a retrofit?

** Is this similar to the system that out there already?>

Yes, but it is the most significant ICBM modernization in 25 years. It's going to be like night and day. There's

going to be a lot more automated processing, versus the manual processing that we do now. It's a significant change for us.

** Are you doing both Minuteman and Peacekeeper?>

That's right. REACT was supposed to go into four types of systems; Minuteman, Peacekeeper-Minuteman Silos, Small ICBM, and Peacekeeper Rail Garrison. There were some constraints. The console can only be X because they wanted it to fit on a rail car. Now, we probably won't get Rail Garrison, but we still have a console that fits in a rail car. The modernization across the whole ICBM world will probably be the same.

** They're all being done concurrently?>

Right. The things we're going to do is because there might be some base closures. That has affected our production decision, for number of buys. After the smoke clears, the missile bases that are out there are the ones that we're going to upgrade.

** How do you think it could have been done better?>

Two things. The integration and good requirement analysis done at the beginning. In long range planning, there has to be more involvement of the user to say, if I'm going to get a program, then give me some lead time. Give me the opportunity to have specific blocks to build on. If I don't make the next block, if I don't go into the next phase, at least I've got a completed phase that I can shelve

and bring off the shelf later on if I deem so, and upgrade it with new technology or whatever. A lot of times we find ourselves in 2 or 3 different phases, and then they shelve us right now, we're about to start production, we're in about the middle to end of Engineering/Manufacturing phase. Let me finish my engineering development, then I make my decision. But, because of timing, I've got to do a lot of this stuff in parallel. If the Government hits, we're ahead of the program. If we don't, we've lost some critical time that these guys said absolutely couldn't be lost at the beginning, or I would have had better requirements definition or something.

Mission Planning Program Manager

Phase I was developed for a separate purpose. We're talking operational capability, so we maintain that threat throughout the history. The purpose was to load data that's planned somewhere else on the TRIAD computer system, on to the ALCM on board the B-52G. That was in the nuclear role for that missile. Historically, SAC aircraft are data intensive. This is a multi-phase aircraft, with a very long range mission, carries a ton of missiles. Guided missiles means you launch them and forget them. They have their own navigation capability. You don't have to work them with a joy stick. It's not like a laser guided bomb that you have to be able to see an image out of a seeker. these are completely autonomous of the aircraft when it flies. Because of that, we're very data intensive. So, all of SAC nuclear planning has been data intensive since the beginning.

First thing we did was SMDPS Phase 1 to program ALCMs. As the avionics on the B-52 became more sophisticated, you needed a way to also load the nuke data to the aircraft so it could get to the launch point. Plus new weapons were being added like the SRAM. The only way of getting data in there is to have some human being physically key in the data, or else you turn around and load via computer. SAC opted to load via computer, hence Phase 2 was born from Phase 1. So, we started out with Phase 1 with the ALCM.

Phase 2 then was SRAM A, plus the capability to program the on board flight avionics. Also for the B-1 to program the defensive avionics suite on board the airplane. So, B-52 was more or less of an active defensive system on board the aircraft. The B-1 came on board and needed a whole lot more information to be able to do the defensive job; more than the B-52 did. So, for those reasons they developed the Phase 2 capability.

Then they were going to transition to a Phase 3 capability. They perceived their need at that time to have an interactive capability so that someone could sit down at the unit and plan a mission. Plan the departure and recovery routes. The only thing that the TRIAD computer plans is the strike portion of the mission. When you coast into hostile territory, the TRIAD computer programs that. However, getting to that point would have to be planned manually. They wondered why you couldn't do that on a computer. When you're done with the strike mission and you've exited the target, all that part would have to be planned manually. SAC a way that they could plan those automatically. Additionally, we wanted to be able to generate mission materials. They had a requirement under the Phase 3 program to do two things: develop interactive route planning, and they also wanted a way that they could generate crew mission materials. They went out and said, what we're going to do is get a cadre of 25 blue suiters.

We're going to ask Systems Command to rent a building in Omaha, Nebraska to house those people. Out of the program line we're going to provide 75 contractors to support these efforts. Since SAC did not have any program management expertise, they also asked Systems Command to hire the BMAC to act as program manager for all these independent contractors. These contractors did not have product-oriented deliveries. It was sort of like a personal services to a degree. Everybody was bringing in their own expertise. They had associate contractor agreements, but with no specific deliverable envisioned. It was up to the blue suiters to manage this total effort. McDonnell Douglas would be doing most of the work. They also had to interface with the contractor doing the Combat Mission Folder, which was Logicon. Additionally, Logicon did IV&V. But there was no deliverable. They were just providing expertise to this group. As a result of this loose structure and lack of focus on deliverables, after the first 18 months of performance the effort schedule had slipped 34 months.

Headquarters SAC asked for help from Systems Command. We went in and we baselined that program in late 1989 or early 1990. We came back with a report to SAC; the constraints we're under were how can we fix this program from a management point of view. How can we make that Phase 3 concept, where we're developing all the pieces at one time, work. We came back with five alternatives to SAC.

The lowest cost alternative was to take some of the difficult work away from the blue suiters, away from that non-descript group of contractors, package it up, do a source selection and give it to one contractor with total system performance responsibility. He would build the hard piece. He would also have the responsibility of integrating the Government-developed easy piece. We also allocated some of the easy tasks to the blue suit programmers. We suggested that SAC do away with that group of contractors that didn't have a product focus. SAC looked at that and said, well, we like everything you've told us, except for the part about having the blue suiters do it. You've convinced us that we are not a software development organization.

They asked us to come back with another alternative. That alternative had to be low cost, because of all the cuts they were receiving in funding. It had to deliver the existing capability. What they really wanted was a deployable capability. They wanted to be able to take the unit level data preparation capability and, in times of heightened conflict, move that capability to another operating location. We went from Phase 1, which was just a cruise missile planner, to Phase 2, which brought in the capability to put data in an aircraft, to Phase 3, which was envisioned to be an interactive, state-of-the-art planner to cut mission materials; to do a much bigger job, plus to be

deployable. SAC then reneged on that requirement and said, right now the first thing we really need is that deployability. We want you to take the Phase 2 system, continue that, but add to that Phase 2 system the deployable capability. That became our NMPPS Block 1. Then they gave us another list of generic capabilities that defined their goals. Everything that was a first priority goal was put into the Block 2 system. In order to enhance the responsiveness and flexibility of the bomber force, SAC wanted a capability that they could crank out those crew mission materials in a relatively responsive way. We are in the process now of developing the Block 2 capability for them. What that will do is just do printed charts. It's the first time that we developed a system that would be responsive to all the SAC aircraft. It would do charts for B-1, B-2, B-52, KC-135 and KC-10.

** So, You're not on contract for either of those efforts?>

No, we are in the final throws of getting on contract. We set up a competitive RFP on Block 1 on 18 January. We received our response on 27 February. There was only one response. We thought we had a competitive environment, but the two competitors teamed. What that does is, under normal source selection the effect of competition sharpens the offerors pencils. So, you have responsive proposals. The level of detail, by regulation, is not required to be as great as far as how well you look. When you go into a sole

source environment the FAR requires you to evaluate that one offeror more closely to be sure that the Government is receiving a fair and reasonable offer. We needed some detailed information that the contractors were not accustomed to giving the Government. It's taken a while. In order to minimize any impact on the operational capability to SAC on those yearly tapes, we used our old contract to do a very small part of the work for the new contract since it's going back to the same fellows. That's helped blunt the impact of the delay; but we're still trying to get on contract.

On the Block 2 program we're in the requirements definition phase. We have working groups out with the user that have taken their SORD, that defines the entire nuclear program, and decomposed those requirements so that we can see in greater detail what, not only operational capabilities, but the performance requirements that they have. We've been directed by SAC management to actually question requirements, and to scrub them to make sure that the requirements in the SORD are in fact valid. And to do a scrub of the requirements versus existing and projected technology. SAC doesn't want us to design a high dollar value thing into the proposal without checking with them first to make sure that it's a got to have, and to make sure that there's no manual way around it. One example, in the Block 2 program is, one of the time consuming things after

you've printed out all the charts is simply to take the charts and put them in plastic sleeves that form the crew mission folders. The initial requirement was that the entire book would be built in an automated way. When we started to look at that, we found out that to design the device to be able to do that would have been a very costly effort. That's going to remain a manual process, at least for the near term. Block 3, we've reviewed the SORD requirements, but we're not in active planning for that because there is no resource base, no money to support the development.

** What do you think was the reason for the 34 month slip in SMDPS Phase 2?>

It was a function of a couple of things. Number one - we had an organization that had never done a software development. You can have the smartest programmer in the world and put in as program manager in a software development effort, and he'll fail miserably. The reason is because there is a lot of integrated specialties, and there's a lot of testing events that have to occur. Second of all, the reason that there was such a great schedule slide was that there was no programmatic focus on developing realistic schedules. SAC management says, I need the new capability by November 1991. This was starting out in '88. Nobody did a serious study on how long it would take to develop that code. Nobody had done a requirements analysis

to find out how to do it. So, they worked on that delivery date, and worked backwards.

The user had very poor requirements discipline. One of the reasons the user wanted to do this is that the user perceived this would give him a tremendous amount of requirements flexibility. Instead of having a formalized requirements structure, what was happening under the SAC management is the DO shop had direct access to the developers without going through the program management office. They would be changing elemental requirements without a formal change in the baseline. As a result you get top level design, then there would be some change in requirements that management wasn't aware of, and then you'd go back to top level design again. You never got out of that loop, so you never progressed to the actual software design. That was the problem. One of the things that we did to prevent that was we formalized our relationship with SAC. The user has a mouthpiece at the headquarters. We've looked at SAC/XR, who is in fact the requirements focal point in SAC. We don't respond directly to the operation user, although we do dialog with them. Whenever there is a change in requirements, we respond directly to SAC/XR, which is the way it should be. That was a way to put some bounds on requirement creep.

** What was the reason for the different tapes on the SMDPS program?>

They did two things: 1) adding incremental capabilities that does not significantly alter the characteristic of the acquisition plan. And 2) maintaining existing software, resolving discrepancies. The effort every year finds some bugs in the software. Also, if there's a change in the aircraft itself, or a change in the central planner at the Headquarters level, my mission planning system has to mirror that change. Part of the enhancements to the mission planning systems are actually efforts to just keep pace with developments external to mission planning. Then, on a rare occasion, you'll get a new weapon on the B-52 or B-1B, such as SRAM II. There will an implementation of SRAM II on TRICOMS, and it will have to pass through the pipeline out of TRICOMS into the unit level planner. I'll have to make a change to accept the data on this end, reformat it, and put it on the DTUC that my machine makes, and then be ready to plug it into the airplane. Without mirroring that capability, there's no way that the missile itself can launch. Is the work maintenance or is it enhancement, or is it new development? It's a very gray area. It depends on what you're talking about. Since the funding appropriation is such, right now it's O&M money, generally we characterize the efforts as operations and maintenance.

**** Is that same software being used for NMPPS?>**

Yes, on a new piece of hardware thought to perform better, also thought to be more logistically supportable.

Our last effort on the old contract was to do a special study, because SAC has now asked us the question, is there a way to minimize risk? We were given an implementation from SAC saying, deliver the same functionality I have today, in a year, on a new system. About the only way you can do that is to deliver the same software on a piece of hardware that's compatible with the old hardware. And what we're doing is upgrading the peripherals. Part of the study is to look at risk minimizing, because that implementation forces us to go to a company called Concurrent Computer. Just to make sure the operational capability is there, we've been doing some risk minimization studies. We're going to present the results to SAC on or about the first of August, to tell them that you could do it the way we scoped it out, here's the schedule, and here's the pros and cons. SAC may look at our alternatives that we're going to give them and say the 75% solution, at a significantly lower cost and greater flexibility, may be the way to go. We may have a change in the program baseline. But, it would probably be similar to the program that we've got now in terms of ops concept.

** You said that 95% of the effort was software?>

Yes, I think as you focus on computer development you'll find that the thrust in general now is to buy COTS hardware. If you have a deployability requirement, then you do something to modify the COTS. I think the military is

getting out of where you develop your own hardware. Especially for data processing applications, because the environment is so rich in the commercial sector. Capabilities are so varied that you have a good field to pick from as far as COTS hardware. So, I don't go out and develop hardware, I use industry standard hardware to run my development software on. You get a different type of program, like DFMR, where they're developing radio systems, and they get into the hardware manufacturing end. But not me, I develop data processing systems.

** Does tape 8 meet all of its requirements?>

Our requirement is to optimize the use of the aircraft. I get new requirements every year. There was never a SORD that was stated in detailed terms that tape 8.1 would have these functional requirements. As a matter of fact, the requirements definition for tape 8.1 actually was done in mid October of last year, about five months prior to tape delivery. The requirements communication process, we called the MIP program, the user takes a look at the tape he's got, and says these are the capabilities I've got to have. They pass those requests to us via either Software Discrepancy reports or material Improvement Requests.

As far as a Baseline Correlation Matrix, I think that's what you're trying to speak to, no there isn't. Because, every year we look at the submitted MIPs and we try to check off that these have been incorporated, or these have been

carried forward because there was not enough available effort in this fiscal year. SACs primary criteria is delivery of the tape on time with new capabilities. There is no set of minimum capabilities. Our requirements base is somewhat flexible and evolving.

**** Why were Block 1 and Block 2 divided up?>**

A couple of different things. the Block 1 effort was going from SMDPS Phase 2, we changed the name to Block 1 and added a deployable capability. But you're essentially using the same operational concept, the same software, and the same overall functionality of the system. Also, we fund that with operations money, 3600 money.

The Block 2 program, on the other hand, is a very clear interface. The Block 2 program was going to plug in to the Block 1 program. It's not going to be a software module; the Block 2 program is a work station. It can work with the Block 1 program for the B-1 and the B-52, it can work with the B-2 SMDPS program to support the B-2 requirements, to work with Busy Planner to support the B-52 conventional capabilities if requested, it's also going to interface with the KC-135/KC-10 lap top planner. It's going to be able to cut a mission materials package for KC-10, KC-135 and RC-135.

**** Block 2 lags behind Block 1, is that to get some learning?>**

No, completely independent in development. No learning transfer at all. Since Block 1 is an evolution of the SMDPS Phase 2 program. That does mission data preparation, that's an operational capability. The Block 2 prints charts. It uses some data from the Block 1 system, but it's like a parasitic relationship. Really, they're autonomous systems. You don't have to have an NMPPS Block 1 to have an NMPPS Block 2 in the squadron. As a matter of fact, you may have an NMPPS Block 2 in the squadron and a KC-10 lap top planner. You plan your mission on the KC-10 laptop planner, you cut your mission data set, then you take this over to your Block 2 workstation. You plug it in, it would read the disk, and it would say this is a KC-10 mission. The system itself is rule-based. It knows the types of charts I need, the number of charts I need...it just cranks them out. I don't have to interactively do anything with it because it knows. Just on the route of flight, it can do that work. Current systems in the field that can do similar capabilities require a lot of manpower. When you look at the size of the effort for a strike mission, that's 600 printed charts.

** How many software people do you have?>

Our systems engineering is done by the MITRE Corporation. Allocated to this program in terms of strict systems engineering support is 44 people. Those 44 people are divided in the four programs. There's not an equitable

allocation. There's roughly 6 working CMPPS, 15 working NMPPS Block 2, and 8 working NMPPS Block 1.

** Is there a lot of software background in those people?>

No, because we're actually an autonomous SPO to a degree. We have our own matrix support within the organization. Some of the people are data and config specialists; data and config in the sense of software acquisition. Where we have knowledgeable people on software is in each of the programs. Some program managers have knowledge in software, but they're backed up also with a deputy that has considerable strength in programmatics. I have a dedicated software force that has a detailed expertise. We have program managers with varying degrees of software familiarity.

** Do you think there has been improvement between increments?>

What I think is, is the Air Force getting smarter in mission planning, or is more focus in it? Historically, as users became more computer literate everybody found that data processing was a way to improve productivity, so everybody went off and developed one. Most of the major Commands had acquisition arms that they could go out where they were very close to the user and actually go out and design systems that were extremely user specific. The problem was not that the software was not compatible. The problem was the data that we need to manipulate. As you

develop your own ADP system, you wanted to be able to swap data. As a community, we did not focus on what was the job we wanted to do first. We did not have the joint perspective. We didn't have the interchange perspective. Then we started to look at hardware capabilities first. IBM just came out with the new workstation. We can do a lot of good stuff with a workstation, because it's a great user interface. You take your hardware and say, what kind of operating system does it support? What kind of DBMS works on that operating system? Finally, what kind of data can this DBMS handle? I manipulate the nature of my job to follow what I've been able to procure, instead of really taking a look at the nature of my job in terms of what I want to do. The very last thing that should have been considered was the kind of hardware to put this on. If you do it that way, you become less dependent on the hardware, and you become more focused on data. Data is a major investment. What gets cheaper is hardware. Hardware gets cheaper, and it does more as time progresses. If you focus at data, and the last thing you consider is hardware, you're going to look for an operating system that is more or less compatible, that focuses through a whole spectrum of hardware implementations. You know that they are going to have considerable longevity. That gives the message that if the military focuses on the software side, it gives the developers the message, you better build something that is

software compatible with other stuff that I've done before. If we start with hardware-based implementation, then you turn around and say, now I would like to upgrade this hardware and make it more logistically supportable or make it work faster. So, you change the hardware, but because the hardware drove all those other things, you have to go down and redo all your software, which in essence makes you make another considerable high cost investment of redoing your data. So, actually then software holds you back from your hardware evolution. Where as if you make it a very standard software package, it really doesn't care what hardware you've got it on. You can upgrade your hardware pretty easy. That's the business that we're in right now. I'm trying to migrate my old system that was based on a hardware solution, to open system architecture solutions. New developments that I'm doing, like the Block 2 program, are going to be based on industry open system architecture standards.

**** Are you going to have one delivery?>**

Two versions. The purpose of the first version is to get it out and into the user's hands. One of the things about incremental deliveries is that it minimizes risk to the users, because you're going to be able to determine functionalities, you're going to get quicker feedback from the user, and you fund in smaller blocks. As you deliver one product, even if you get a funding cut, you've still

delivered something to the field. As opposed to working for one big chunk at the very end; you would not get that opportunity for user feedback and, if you have a funding cut at a critical part, then you've delivered nothing.

Everything you've done is completely wasted.

** What are the political advantages of incremental deliveries?>

When you consider delivering a product incrementally, what you're really trying to do is -- I was given this finite pile of money, the best way that I can do it and galvanize realistic contractor response is don't give them too big a piece of pie to chew on. Give them incremental pieces of the pie to chew on. Deliver a capability out in the field, so in worse case you're going to have some significant milestones, shorter in, to be able to assess success. You're going to be able to have a leveler type of effort, where you're incrementally testing and in parallel doing concurrent development on a second version. You have a tendency to level out manpower, you have a tendency to decrease risk because a guy that can develop code can very often test code. The same kind of human being, same level of expertise, same area of interests. You can take that delta in manpower and move it on to the second version that should be ramping up. Pretty soon you're able to deliver a capability, it's a more sustained effort. it's an effort

that the contractor perceives as one of less risk, and you can amortize a fixed overhead at a more realistic rate.

** Does it help estimate what the costs of the next version are going to be?>

Sure, it minimizes risk. It gets something out in the field. The big problem for any program manager is the fact that our resource base changes radically. The goal is to maintain stability. I can deliver a better product, even if you cut me back in dollars. If you can start me out at a lower dollar figure, I'll deliver you my best shot. Just promise me that we'll stay constant.

** What do you think would happen if it was done traditionally?>

I think it's more costly. I think there's more risk in terms of programmatic, because historically funding levels aren't level. If you turn around and say just one delivery at the end, there are so many nodes on your critical path that any one of which could break down, and you get the reputation that I never delivered the product. If you do an incremental delivery, I have certain requirements I'm trying to meet in version 1, certain requirements in version 2; there's some flexibility. If I get a funding cut, I'm allocating efforts to certain operational capabilities in version 1, I can say I'll give you your top three in version 1, I have enough to do that. Then I can slip others to version 2 and press on. So, I can still deliver the

capability by versions. But, if I deliver it in one fall swoop, there's a lot more interrelations with functionalities. Instead of doing multiple 2167A milestones, you just do one big one. If I have a change to the baseline, it's one big one.

** Do you think that one time through would work for simpler programs?>

Once you're going with a contractor, contractor surveillance is not difficult. It requires good people, but it isn't brain surgery. The hard thing to do is get something on contract that is a fair deal.

Program Control Officer

GWEN is incremental in the sense of quantity. the first phase consisted of forty transmit stations scattered throughout the country, and that would grow to bring it up to around 100. Politics are what got in the way of that program. It's on hold until congress evaluates the results of a study as to what the health hazards are of low-level radiation. the study will be done at the end of this year, and Congress has a year to evaluate the study. So, we wont be doing anything on that program until '93.

** Do you think there were any benefits to getting a core capability out there early?>

Yes, the Thin Line Capability was done during R&D. The expansion network was production. All the units produced are in storage. All we're waiting for is permission to install the equipment at the sites. A lot of the sites have been selected too, but the Congress put a hold on construction, so we just can't do anything. That's incremental in a sense that you go from a minimum capability to a full capability.

** Did the contractor perform better going from R&D to production?>

GE was the development contractor. My impression was that they performed adequately from a technical point of view, but there might have been some financial problems. If I'm not mistaken, they overran. Typically, in a fixed price

type contract there are overruns, so I would think that there was an overrun on the R&D phase of this program. Then in production, it was fixed price also, but I don't think there was a cost problem.

** Maybe they got a little smarter after doing the first increment?>

GE produced the system, and then Contel, who's the installation contractor, they're performing well from the information I've got.

** That wasn't broken up for any funding reasons was it?>

No, that was the original plan. The Army Corps of Engineers is responsible for acquiring the land. That's a whole separate project, and it is very complicated dealing with land owners and trying to lease and purchase sites.

MISSION PLANNING

That's definitely an incremental program. The first block is basically to transfer what exists now to a new set of hardware. After that there will be growth in capability.

** Have they selected what the new hardware is going to be?>

That's in negotiation right now, but we know what the hardware will be. It's a Concurrent Computer, and will be off the shelf.

** Is SMDPS still going?>

This is another name for the same thing. The Combat Mission Folder part means that the computer generates maps,

compass bearings and altitudes. It automatically comes out of the computer. the pilot will take the folder along with him to the cockpit for the mission.

** Did the existing system help the user understand the requirements up front?>

We've only been working on this project for a year. Prior to that SAC was doing all the work in house. They were doing all the software development. I think they were trying to do it in one big block, which we have broken down into three blocks.

** What three blocks are you talking about?>

The first one being the translation of software from existing hardware to the Concurrent computer. The next phase would be the Combat Mission Folder. And the third phase is the ultimate system, which involves even newer hardware, state of the art software, and very automated. I don't know if that has been defined at this stage. The definition is fluid, and they keep changing.

** Does phase 2 lag behind phase 1?>

Yes, but there will be some overlap. Before the final phase 1 system is installed, we'll be starting phase 2.

** Do you think it is a good idea to wait on phase 2 instead of doing them concurrently?>

Yes, I think so, because you only have so many people, and you can't do everything.

REACT

There's different aspects. We're involved with the communications, command and control aspects. BMO is involved with WSCE.

** Is that the same contract?>

No, they're different contracts. The BMO contractor is Ford.

** What systems is REACT going to interface with, just Minuteman?>

(Julie) There is a possible interface with the Peacekeeper missile. It's an option on the contract. It's not being exercised. We're looking at interfacing with DFMR, and Small ICBM.

** I've read the management Assessment Reports, and they've mentioned a 50% memory reserve problem, what is that?>

That's a big problem on REACT. That's one of the stumbling blocks on the contractors. After all the software is developed and working there's supposed to be 50% extra capacity for future growth, and the way GTE has approached the situation to date, they don't have that kind of capacity. Technically, they're not in compliance with the contract. They could solve this by redesigning the software or providing more memory. But those are things that will cost them money, and they're in a loss position. So, I think they're trying to not solve the problem and find some

way around it. There are two or three areas where there are technical hangups on the REACT program.

** And there's a 40% cost variance.>

That means that they're in an overrun state, and that's why they don't want to invest any more company money in the program. It's difficult for us to insist that they give us the 50% memory capacity when it's going to cost them money.

** It sounded like the reason for the cost variance was the manhours needed for requirements derivation, is this true?>

When they first started, in order to come to grips with the requirements and basic design, they had to put people on that were much more talented than they really wanted to, because of the complexity of the program. So, the cost of people, the high-priced engineers, was more costly than they would have liked. Plus they had to work more. Like rather than 10 systems engineers, they had 20 for a longer period of time. They had more expensive people and more of them, and they worked them longer than they expected.

** Did you think it would have helped if they broke it down into smaller pieces?>

The hardware is basically off the shelf. There might be some modification for nuclear hardening. I don't think the problems they have are what you call show stoppers. I think they're workable. It's just a matter of how you solve it. Which is the cheaper solution, as opposed to gee, this is something we can't figure out. I think basically the REACT

is a successful program. It's just difficult because the company is loosing money, and they're trying to find the cheapest way out as opposed to doing it the way the Government wants them to.

** It seems like it was tougher than they thought, is that true?>

Yes, I think that's true. It's like a lot of programs -- when you're competitive during source selection, you try to low-ball it. You try to minimize to get the contract, hoping to recoup on production. Then you find out when you get into it that you may have underestimated the complexity, so you end up losing money. That's why most of the new development programs are cost plus. You shift the risk away from the contractor to the Government. Whatever the contractor spends he gets reimbursed for. So, the risk is on the Government. Up until a couple of years ago, we tended to go fixed price contracts. we put the risk on the contractor, and if he's under pressure to bid low to get the contract, he's liable to underestimate. I think that is why GTE is behind cost wise, they just underbid or didn't understand the complexity, or a combination of things. I don't think the problems that they are facing are unsolvable, it's just that they're trying to conserve dollars.

** Is there any schedule variance along with the cost variance?>

The schedule variance was pretty much in line. It might have been out 8 - 10%, nothing significant. I don't believe any major milestones have been slipped. CDR was a little longer than it should have been. The CDR itself was pretty much successful.

Bibliography

1. Awani, Alfred O. Data Processing Project Management. Princeton NJ: Petrocelli Books, 1986.
2. Benbaset, Izak, David K. Goldstein, and Melissa Mead. "The Case Research Strategy in Studies of Information Systems," MIS Quarterly, September 1987, p. 369-386.
3. Brooks, Frederick P. Jr. "No Silver Bullet: Essence and Accidents of Software Engineering," IEEE Computer, April 1987, p. 10-19.
4. Davis, Alan M., Edward H. Bersoff and Edward R. Comer. "A Strategy for Comparing Alternative Software Development Life Cycle Models," IEEE Transactions on Software Engineering. October 1988, p.1453-60.
5. Defense Systems Management College. Joint Logistics Commander's Guidance for the Use of an Evolutionary Acquisition Strategy in Acquiring Command and Control Systems. Fort Belvoir, March 1987.
6. Department of the Air Force. Lifecycle Management of Computer Resources in Systems. AFR 800-14. Washington: HQ USAF, 29 September 1986.
7. Department of Defense. Defense System Software Development. DOD-STD-2167A. Washington: Government Printing Office, 29 February 1988.
8. Gilb, Tom. Principles of Software Engineering Management. New York: Addison - Wesley Publishing Company, 1988.
9. Glaser, Barney G. and Anselm L. Strauss. The Discovery of Grounded Theory: Strategies for Qualitative Research. New York: Aldine De Gruyter, 1967.
10. Gremillion, Lee L. and Philip Pyburn. "Breaking the Systems Development Bottleneck." Harvard Business Review, March-April 1983, p. 130-7.
11. Lincoln, Yvonna S. and Egon G. Guba. Naturalistic Inquiry. Beverly Hills: Sage Publications, 1985.
12. Metzger, Philip W. Managing a Programming Project. Englewood Cliffs NJ: Prentice-Hall, 1981.

13. National Research Council. Adapting Software Development Policies to Modern Technology. Washington DC: National Academy Press, 1989.
14. Paschall, Lee M. Air and Satellite Communications. Washington DC: AFCEA International Press, 1985.
15. Price, Stan. Managing Computer Projects. New York: John Wiley & Sons, 1986.
16. Roetzheim, William H. Structured Computer Project Management. Englewood Cliffs NJ: Prentice Hall, 1988.
17. Schatzman, Leonard and Anselm L. Strauss. Field Research: Strategies for a Natural Sociology. Englewood Cliffs NJ: Prentice-Hall, 1973.
18. Stone, E. Research Methods in Organizational Behavior. Scott, Foresman and Co., 1978.
19. Yin, Robert K. Case Study Research: Design and Methods. Beverly Hills: SAGE Publications, 1984.
20. Yourdon, Edward. Managing the System Life Cycle. Englewood Cliffs NJ: Yourdon Press, 1988.

Vita

Captain Martin W. Wituszynski was born on 26 August 1962 in Syracuse, New York. He graduated from Liverpool High School in Liverpool, New York in 1980 and attended the U. S. Air Force Academy, graduating with a Bachelor of Science in Aeronautical Engineering in May 1985. Upon graduation, he received a regular commission in the USAF. He began his acquisition career as a Configuration Manager in the Strategic Systems System Program Office at Aeronautical Systems Division where he managed the engineering specifications for B-52 and FB-111 subsystems. He was then chosen to serve as the test manager for the Short Range Attack Missile (SRAM) II. There he was responsible for the development testing of the SRAM II on board the B-1B and the F-15E until entering the School of Systems and Logistics, Air Force Institute of Technology, in May 1990.

Permanent Address: 105 Timber Lane
Brownsburg, IN 46112

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Project, (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE September 1991	3. REPORT TYPE AND DATES COVERED Master's Thesis		
4. TITLE AND SUBTITLE SELECTION OF A DEVELOPMENT METHODOLOGY FOR THE ACQUISITION OF COMMAND, CONTROL AND COMMUNICATION SYSTEMS		5. FUNDING NUMBERS		
6. AUTHOR(S) Martin W. Wituszynski, Captain, USAF				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology, WPAFB OH 45433-6583		8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/GSM/LSY/91S-26		
9. SPONSORING MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSORING MONITORING AGENCY REPORT NUMBER		
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION AVAILABILITY STATEMENT Approved for Public Release; distribution unlimited		12b. DISTRIBUTION CODE		
13. ABSTRACT (Maximum 200 words) This study investigated command and control projects to see if certain program characteristics help determine the appropriate development methodology. A literature search revealed no previous research in this area; but did identify at least two possible development methodologies, and some suggestions for key characteristics. Four case studies were conducted to study the effects of certain characteristics on the success of the development methodology employed; either conventional or incremental. Requirements definition, availability of a previous system, system complexity, number of subsystems, development experience, user involvement, and funding all appear to impact the appropriateness of the two methodologies investigated.				
14. SUBJECT TERMS Command and Control, Acquisition, Development, Management		15. NUMBER OF PAGES 138		
		16. PRICE CODE		
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL	

AFIT RESEARCH ASSESSMENT

The purpose of this questionnaire is to determine the potential for current and future applications of AFIT thesis research. Please return completed questionnaires to: AFIT/LSC, Wright-Patterson AFB OH 45433-6583.

1. Did this research contribute to a current research project?

- a. Yes b. No

2. Do you believe this research topic is significant enough that it would have been researched (or contracted) by your organization or another agency if AFIT had not researched it?

- a. Yes b. No

3. The benefits of AFIT research can often be expressed by the equivalent value that your agency received by virtue of AFIT performing the research. Please estimate what this research would have cost in terms of manpower and/or dollars if it had been accomplished under contract or if it had been done in-house.

Man Years _____ \$ _____

4. Often it is not possible to attach equivalent dollar values to research, although the results of the research may, in fact, be important. Whether or not you were able to establish an equivalent value for this research (3 above), what is your estimate of its significance?

- a. Highly Significant b. Significant c. Slightly Significant d. Of No Significance

5. Comments

Name and Grade

Organization

Position or Title

Address