NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA

THESIS

**VALUE ENGINEERING:
AN APPLICATION TO
COMPUTER SOFTWARE**

by

Charles T. Race

June 1995

Thesis Advisor: Jeffery Warmington

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE June 1995	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE VALUE ENGINEERING: AN APPLICATION TO COMPUTER SOFTWARE	5. FUNDING NUMBERS
6. AUTHOR(S) Race, Charles T.	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey CA 93943-5000	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER

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

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

13. ABSTRACT *(maximum 200 words)*

The purpose of this thesis is to determine how and to what extent can the Department of the Department of the Navy's Value Engineering Program be utilized in the acquisition of computer software. A review of professional literature such as journals, periodicals, and research reports provide the background information necessary to explain potential relationships between Value Engineering (VE) and computer software. Surveys were submitted to all Department of Defense (DOD) Program Managers, U.S. Navy Systems Commands, and Defense Contract Management Command Districts to determine how senior DOD management currently perceives the VE computer software relationship. An analysis of the data resulted in the following conclusions: (1) the Federal Acquisition Regulation part 48 does apply to software, however, it was written with an emphasis on hardware and unit cost reduction; (2) the methodologies of VE do apply to computer software development and acquisition; (3) DOD software acquisition policies do not effectively support the utilization of VE; and (4) contracting personnel and Program Managers require additional training in software development. Value Engineering is an effective contracting tool that can offer tremendous opportunities for Government and industry alike when used appropriately.

14. SUBJECT TERMS Value Engineering, Software, and Reuse	15. NUMBER OF PAGES 92
	16. PRICE CODE

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

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18 298-102

Approved for public release; distribution is unlimited

VALUE ENGINEERING: AN APPLICATION TO COMPUTER SOFTWARE

Charles T. Race
Lieutenant, United States Navy
B.S., State University of New York at Cortland, 1985

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN MANAGMENT

from the

NAVAL POSTGRADUATE SCHOOL
June 1995

Author: Charles T. Race
Charles T. Race

Approved by: Jeffery Warmington
Jeffery Warmington, Principal Advisor

David V. Lamm
David V. Lamm, Associate Advisor

David R. Whipple, Chairman
Department of Systems Management

ABSTRACT

The purpose of this thesis is to determine how and to what extent can the Department of the Department of the Navy's Value Engineering Program be utilized in the acquisition of computer software. A review of professional literature such as journals, periodicals, and research reports provide the background information necessary to explain potential relationships between Value Engineering (VE) and computer software. Surveys were submitted to all Department of Defense (DOD) Program Managers, U.S. Navy Systems Commands, and Defense Contract Management Command Districts to determine how senior DOD management currently perceives the VE computer software relationship. An analysis of the data resulted in the following conclusions: (1) the Federal Acquisition Regulation part 48 does apply to software, however, it was written with an emphasis on hardware and unit cost reduction; (2) the methodologies of VE do apply to computer software development and acquisition; (3) DOD software acquisition policies do not effectively support the utilization of VE; and (4) contracting personnel and Program Managers require additional training in software development. Value Engineering is an effective contracting tool that can offer tremendous opportunities for Government and industry alike when used appropriately.

TABLE OF CONTENTS

I. INTRODUCTION 1

A. GENERAL 1

B. OVERVIEW 1

C. RESEARCH OBJECTIVE 4

D. RESEARCH QUESTIONS 4

E. SCOPE OF RESEARCH 5

F. RESEARCH METHODOLOGY 5

G. CHAPTER OUTLINE 6

II. DEPARTMENT OF DEFENSE VALUE ENGINEERING 7

A. INTRODUCTION 7

B. THE BACKGROUND OF VALUE ENGINEERING 7

C. WHERE DOES VALUE ENGINEERING START? 9

D. VALUE ENGINEERING IN DOD CONTRACTS 11

1. The Federal Acquisition Regulation . . . 11

2. Office of Management and Budget Circular No. A-131 19

3. Value Engineering Guidance For DOD and Navy Use 20

E. VALUE ENGINEERING: A NEW DIRECTION 20

1. The Perry Memorandum 20

2. 104th Congress H.R. 719 22

F. A VALUE ENGINEERING SYSTEMS EXAMPLE 23

G. SUMMARY 24

III. SOFTWARE VALUE ENGINEERING APPLICATIONS 25

A. OVERVIEW 25

B. VALUE ENGINEERING CONCEPTS 26

1. Software Development 27

a. Weinberg's Experiment 28

b. Marginal Analysis 31

c. Goldplating 34
d. Legacy Systems 36
e. Software Reuse 39
f. Software Maintenance 41
C. SUMMARY 47

IV. DOD PROGRAM MANAGER AND SOFTWARE ENGINEERING PERCEPTIONS OF VALUE ENGINEERING IN SOFTWARE DEVELOPMENT 49
A. INTRODUCTION 49
B. SURVEY QUESTIONS AND ANALYSIS 49
1. Question one 49
2. Question Two 52
3. Question Three 54
4. Question Four 57
5. Question Five 59
6. Question Six 61
C. SUMMARY 63

V. CONCLUSIONS AND RECOMMENDATIONS 65
A. OVERVIEW 65
B. CONCLUSIONS 66
a. Education 68
b. VECP Avoidance 69
C. RECOMMENDATIONS 70
a. Revise FAR Part 48 71
D. RESEARCH QUESTIONS 71
E. AREAS FOR FURTHER RESEARCH 76

APPENDIX 77

LIST OF REFERENCES 79

INITIAL DISTRIBUTION LIST 83

I. INTRODUCTION

A. GENERAL

The purpose of this thesis is to develop an understanding of the Value Engineering program within the Department of the Navy, to what extent it is currently being utilized, and most importantly, how the concepts of Value Engineering can be applied to computer software procurements. This chapter provides an overview addressing the potential application of Value Engineering to computer software, the underlying issues that have made computer software a critical management concern, and concludes with a brief organization of this research.

B. OVERVIEW

Value Engineering originated during World War II as the result of intense mobilization requirements and the inherent material shortages that were experienced in order to the meet the tremendous demands of the United States' war fighting machine. To relieve the stress of material shortages, substitute materials were utilized to the maximum extent possible provided that the value and functional utility of the end product were not compromised. In 1947, Lawrence Miles developed the methodology or philosophy which became known as Value Engineering. Another term known as Value Analysis is used synonymously with Value Engineering. This research paper will consistently use the term Value Engineering for simplicity and clarity.

In the Department of Defense (DOD), Value Engineering is defined as,

> a systematic effort directed at analyzing the functional requirements of DOD systems, equipment,

> facilities, procedures, and supplies for the purpose of achieving the essential functions at the lowest total cost, consistent with the needed performance, reliability, quality, and maintainability. [Ref. 29: P.1-2]

The general term "procedures" in the definition above lends itself to the application of software. Software development is predominately a procedure intensive process similar to that of a manufacturing process. Development of both software and hardware typically uses processes which consist of a number of structured phases. Value Engineering is directly applicable to a manufacturing process in that any structured process can be changed in such a way as to increase value to both the customer and the manufacturer. The researcher will demonstrate throughout this paper that a software development process can also be changed in such a way so as to achieve additional value to the customer and software manufacturer by applying the methodologies of Value Engineering to the software development and acquisition process.

At this point a definition of "value" is appropriate. The definition of value is: (1) the worth of a thing in money or goods at a certain time, and/or (2) the utility of an item directly or indirectly satisfying a recognized need [Ref. 2: P.23]. The primary emphasis of Value Engineering includes: (1) The identification of costs as unnecessary and (2) The decision making which will eliminate the identified unnecessary cost [Ref. 19: P.vii].

Of particular concern at the Congressional level is the spiraling cost of computer software in the Department of Defense. In the early 1980s, DOD expended less than ten billion dollars annually on software development and support cost. Recently, DOD spent between $24 billion to $32 billion annually. This figure represents approximately ten percent of the Defense budget. However, approximate software expenditures for fiscal year 1994 have reached $42 billion. It is estimated that by the year 2008, software development and support costs will represent 20% of the Defense budget. [Ref. 31: P.1]

RADM Robert M. Moore, while Commander of Naval Information Systems Management Center in March 1993 defined the importance of software by stating that,

> At one time, it was the hardware that supported the mission. Today, the hardware is rather generic, capable of supporting any mission. It is the software that provides the real functionality. [Ref. 23: P.10]

To illustrate how sophisticated weapon systems have become over time, a review of the amount of software code contained in them provides a relative indication. For example, fighter aircraft during the Vietnam era had software systems that contained fewer than 100,000 lines of software code. Today's fighter aircraft can easily contain up to six or seven million lines of code. According to some estimates the ballistic missile defense system or the Strategic Defense Initiative, could have 40 million to 100 million lines of code. [Ref. 31: P.2] Today's weapon systems field impressive technological capabilities that are all software dependent to support mission requirements. However, with the ever increasing demand for high technology weapon systems with additional capabilities to meet and counter new and sophisticated threats to our existing systems, the resulting demands for software advances increase tremendously. Unfortunately, software development for new systems can take up to 10 years or more to develop and within that time threat assessments can and do change which require corresponding changes to the software development. As a result, software costs have spiraled out of control with no relief in sight.

Software acquisition is an integral part of the major weapon system acquisition process. This research will identify how the methodologies and goals of the Value Engineering program can be used to enhance the acquisition process for computer software.

C. RESEARCH OBJECTIVE

The purpose of this thesis is to develop an understanding of how the Department of the Navy manages its Value Engineering program with respect to computer software acquisition, and to what extent the methodologies of VE could be utilized to reduce ever increasing computer software costs. It is the goal of the researcher to provide the means necessary for acquisition personnel to seriously consider using VE methodologies as a tool to incentivize defense contractor performance while reducing overall contract cost and still maintain appropriate levels of "Value". Furthermore, it is hoped that this thesis will provide readers with the information necessary to exploit and incorporate acquisition streamlining in all contractual applications of VE to the maximum extent possible.

D. RESEARCH QUESTIONS

The primary research question is derived from the above research objective and asks: <u>How, and to what extent can the Department of the Navy's Value Engineering Program be utilized in the acquisition of computer software?</u>

The following subsidiary research questions were developed to assist in answering the primary research question:

1. What are the principal features of the U.S. Navy's VE program?

2. What is the role of the Value Engineering Change Proposal (VECP) and how is it applied to VE?

3. What characteristics, if any, of computer software acquisition are most pertinent to the application of VE concepts?

4. How do U.S. Navy contractors and in-house personnel view the concept of VE with regards to computer software acquisition?

5. What approach, if any, should the U.S. Navy use to facilitate the application of VE/VECPs to computer software acquisition?

E. SCOPE OF RESEARCH

This thesis develops an understanding of the U.S. Navy's Value Engineering program and how it can be more successfully applied to the procurement of computer software. This study will apply the concepts of VE to the basic principles of software development and acquisition. It is not within the scope of this study to provide an indepth understanding of software engineering and development. Furthermore, it is assumed that the reader has a basic understanding of acquisition concepts, terminology, as well as the basics of major weapon systems acquisition.

F. RESEARCH METHODOLOGY

The research methodology utilized in this study involved a comprehensive review of current literature and surveys submitted to DOD Program Managers (PM), Defense Contract Management Command personnel, and to personnel at the following: Naval Air Systems Command; Naval Sea Systems Command; and Space and Naval Warfare Systems Command. The literature research included a review of: (1) professional journals and periodicals; (2) research reports published by United States military postgraduate schools; and (3) United States Department of Defense publications. The survey questionnaire is presented in the Appendix.

G. CHAPTER OUTLINE

Chapter I provides an introduction to the thesis in such a way as to construct a framework for the problems of software acquisition. The potential application of Value Engineering to alleviate those problems as a possible solution is suggested. Chapter II discusses the U.S. Navy's current Value Engineering Program and the application of VECPs. It also discusses the contractual provisions as outlined in the Federal Acquisition Regulation (FAR). An outline of various Government authority directives as it applies to Value Engineering is also discussed. Chapter III includes a discussion of Value Engineering and its current/potential application to software acquisition. Unique characteristics of software acquisition are examined. An analysis of Value Engineering methodologies is presented in terms of actual and potential usage of the VECP in the application of computer software acquisition. Chapter IV includes a discussion regarding the challenges of acquisition regarding software in today's military environment. Additionally, the current perceptions of key acquisition/software engineering personnel will be discussed as it applies to this study. Chapter V will address conclusions and recommendations, provide detailed answers to the research questions and suggest additional areas for further research in Value Engineering and computer software acquisition.

II. DEPARTMENT OF DEFENSE VALUE ENGINEERING

A. INTRODUCTION

To develop an understanding of Value Engineering as currently utilized within the DOD, this chapter will first provide a brief background outlining the origin and central themes of Value Engineering. An analysis of current regulations as it applies to Value Engineering in Federal contracting will also be discussed in order to provide the framework necessary to address the research questions. Finally, this chapter will conclude with a discussion of the most current Value Engineering issues that are affecting the way the DOD is responding to increased defense commitments and reduced budgets. An example of a systems type value engineering change will be provided to demonstrate the applicability of Value Engineering to a "system."

B. THE BACKGROUND OF VALUE ENGINEERING

In DOD, Value Engineering applies to hardware, and software; development, production and manufacturing specifications; standards, contract requirements, and other acquisition program documentation; facilities design and construction; and management or organizational systems and processes to improve the resulting products. [Ref. 29] The main objective of Value Engineering is to obtain the same function or performance standard at the lowest cost possible. Value Engineering can be successful, however, if function or utility to the end-user can be increased without absorbing additional costs. Value can be increased by (1) improving the utility of something with no change in cost, (2) retaining the same utility for less cost, or (3) combining improved utility with a decrease in cost. Optimum value is achieved when all criteria are met at the lowest overall cost. [Ref. 29: P.1-4]

Lawrence D. Miles, a General Electric employee was the first to develop the ideas of VE shortly after World War II. His efforts ultimately led to the subject of Value Engineering/Value Analysis. Interestingly enough, Value Engineering is not a rigid science like other engineering disciplines. Mr. Miles defined Value Engineering as,

> A philosophy implemented by the use of a specific set of techniques, a body of knowledge, and a group of learned skills. It is an organized creative approach which has for its purpose the efficient identification of unnecessary cost." [Ref. 19: P.1]

Furthermore, he immediately saw the beneficial implications VE had to offer an organization besides added value and lower costs. Mr. Miles recognized that his "philosophy", if understood correctly and accepted in the organization, affected all the vital branches of an organization such as engineering, manufacturing, marketing, procurement, sales, quality control, and management. He believed it was important for an organization to have its departments share a common cause to champion, which if done correctly, would ultimately further the goals of the organization as a whole. The concept of team work in supporting a common cause, such as attaining the highest levels of value possible in an item, can be the genesis of success for any organization trying to survive in a competitive environment. [Ref. 19]

Mr Miles designed his approach to Value Engineering from a basic prospective. First, he developed three simple steps to accomplish a study in Value Engineering followed by five basic questions to achieve the desired results of enhanced value. The three basic steps are:

(1) Identify the function.

(2) Evaluate the function by comparison.

(3) Cause value alternatives to be developed.

The five basic questions of each Value Engineering study attempts to answer the following:

(1) What is the item?

(2) What does it cost?

(3) What does it do?

(4) What else would do the job?

(5) What would the alternative cost? [Ref. 19: P.14-18]

C. WHERE DOES VALUE ENGINEERING START?

Value Engineering produces the most beneficial results, particularly savings, if applied in the earliest stages of design for a system or equipment. A well thought out Value Engineering program that is implemented in the design stage or "still on the drawing board" will reduce the need to retool production facilities in the future. Costs associated with operations, maintenance, and support elements can also be minimized as a result. [Ref. 17: P.438]

In today's competitive business environment, Value Engineering is becoming a strategic tool to capture market share in order to "provide better customer value for equivalent cost or equivalent customer value for a lower cost [Ref. 7: P.39]." This is accomplished using a relatively new business strategy known as "Target Pricing" and "Target Costing". With Target Pricing/Target Costing, an existing product is re-designed and re-developed with a target price and target cost that will provide some guarantee of success in the market place. Value Engineering is the vehicle that is applied to this re-design/re-development process to achieve the target price and target cost while maintaining maximum value to the customer. The Japanese automotive industry has been very successful in competing with their American counterparts by correctly focusing on effective Value Engineering techniques. [Ref. 17. P.438]

Figure 1 shows the importance of using Value Engineering in the design stages of a product by reviewing the nature of product cost throughout development.

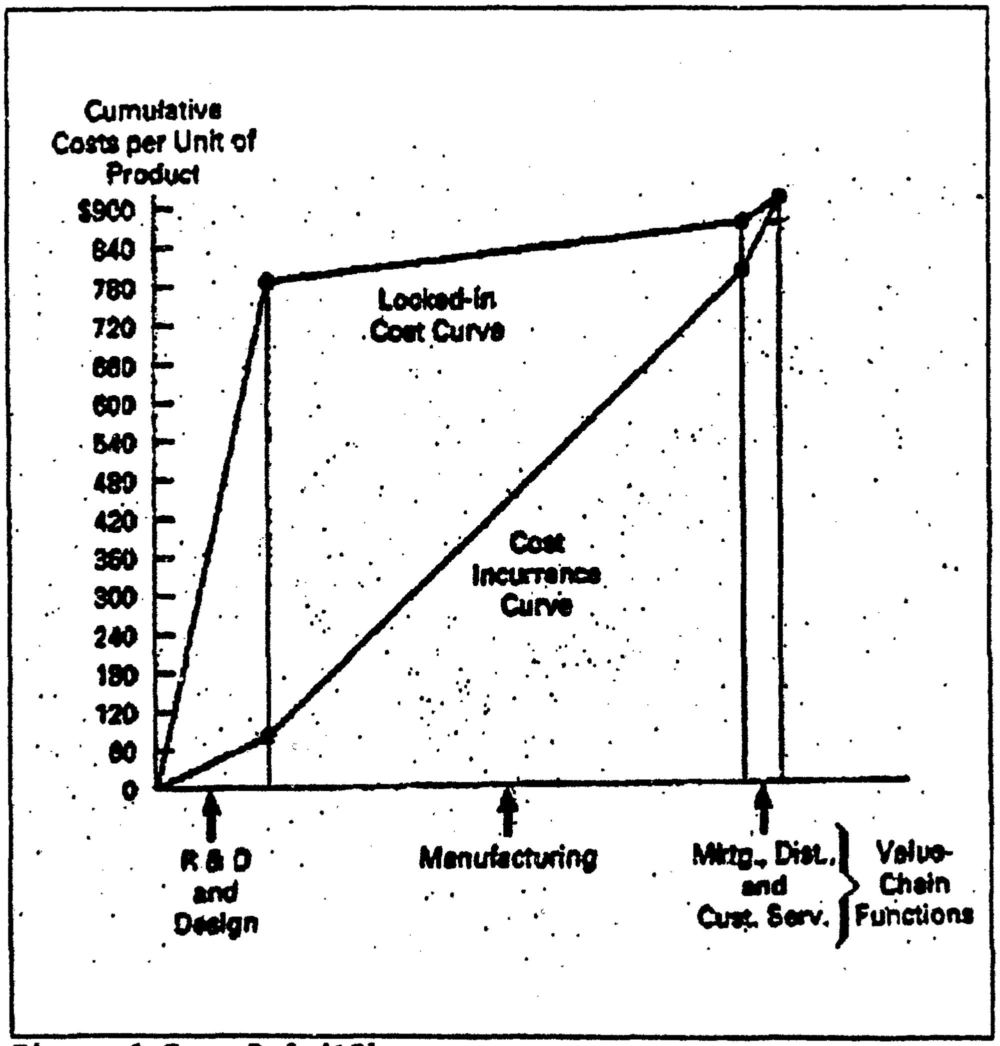

Figure 1 From Ref [17]

Costs are "locked in" because management/technical decisions have been made early in the design stage. These "locked in" costs drive the overall cost of the item throughout the development stage. Cost incurrence are costs that are recognized at the time a cost is incurred during development. It can be seen that locked in costs drive incurred cost because decisions have been made in the design stage. [Ref. 6]

Since a product may be in service for over a century or more, it may well be useful to apply Value Engineering later in the service life of a system or equipment. Specific application requirements of any equipment or system may change over time, whether it is timed in days or years, but the function still remains the same. A good example of this would be the automobile since the manufacturing process in this industry dramatically changes when customers expect more for their money as technology improves.

Regardless of the time-frame involved in the overall service life of a product, Value Engineering should be applied if additional value and profitability will result. Value Engineering studies have resulted in improvements in numerous applications and resulted in:

(1) Service life extension.

(2) Reduced repair costs.

(3) Reduced packaging costs by improving procedures/materials.

(4) Elimination or significant improvement of the function. [Ref. 29: P.2-5]

D. VALUE ENGINEERING IN DOD CONTRACTS

1. The Federal Acquisition Regulation

The main objective of Value Engineering in contracting is to reduce costs while maintaining or improving quality. DOD adheres to the guidance of the Federal Acquisition Regulation (FAR) which discusses the policies and procedures for using

Value Engineering in contracts. Value Engineering clauses are required on acquisition contracts, including subcontracts, exceeding $100,000. The contracting officer may require a Value Engineering clause for contracts under $100,000 if it is believed that potential savings can be achieved. The FAR requires the contracting officer to exempt Value Engineering clauses from the following solicitations and contracts:

(1) For research and development other than full scale development.

(2) For engineering services from not-for-profit or nonprofit organizations.

(3) Providing for product or component improvement, unless the VE incentive application is restricted to areas not covered by provisions for product improvement.

(4) For personal services.

(5) For commercial products that do not involve packaging specifications or other special requirements or specifications.

(6) When the agency head has exempted VE from the contract requirements. [Ref. 11: P.48-2]

Further guidance to assist contracting officers and contractors can be found in MIL-STD-1771A , "Value Engineering Program Requirements Clause".

A Value Engineering Change Proposal (VECP) is a proposal submitted by a contractor, under the provisions of the FAR, that recommends a change in a contract specification, design, or process which would ultimately lower the project's life cycle cost to the Government [Ref. 35]. The contractor submits a VECP through an incentive (voluntary) approach or through a mandatory approach. VECPs approved by the Government which result in contract savings are known as "acquisition savings" and may make the contractor eligible to share a percentage of the savings with the Government through a reduction in the cost of the contract. Acquisition savings include:

(1) Instant contract savings. The net cost reductions on the contract under which the VECP is submitted and accepted, and which are equal to the instant unit cost reduction multiplied by the number of instant contract units affected by the VECP, less the contractor's allowable development and implementation costs [Ref. 11: P.48-1].

(2) Concurrent contract savings. Net reduction in the prices of other contracts that are definitized and ongoing at the time the VECP is accepted [Ref. 35: P.8-1].

(3) Future contract savings. The product of the future unit cost reduction multiplied by the number of future contract units scheduled for delivery during the sharing period. If the instant contract is a multiyear, future contract savings include savings on quantities funded after VECP proposal [Ref. 11: P.48-1].

(4) Collateral Savings. The measurable net reductions resulting from a VECP in the agency's overall projected collateral cost, exclusive of acquisition savings, whether or not the acquisition cost changes [Ref. 11: P.48-1].

(5) Contractor's Development and Implementation Costs. Those costs the contractor incurs on a VECP specifically in developing, testing, preparing, and submitting the VECP, as required by Government acceptance of a VECP [Ref. 11: P.48-1].

Under the incentive approach, the contractor employs his own resources to develop a Value Engineering program and submits VECPs, based on his own efforts, to the contracting officer. This approach is particularly useful since it gives an enterprising contractor the ability to challenge the status quo on his own terms. However, the contractor is reimbursed for allowable development and implementation costs only when the VECP is approved. [Ref. 11: P.48-1]

Under the mandatory approach, a Value Engineering Program

Requirements Clause (VEPRC) is required. The contractor is required to perform a specific level of Value Engineering effort to achieve savings. Under this approach, the Government feels there is sufficient potential to achieve cost savings and may consider the contractor financially incapable or reluctant to perform Value Engineering on their own. Therefore, the Government will pay for or provide the contractor with the necessary resources which will enable a contractor to submit VECPs to the contracting officer. Any cost sharing under the mandatory clause will make the contractor eligible for a lower percentage of the savings, if any, which are otherwise available under the voluntary clause.

The contracting officer makes the determination whether or not a voluntary incentive or mandatory VE clause is required. Under the mandatory clause, the Government incurs additional risks since there is no guarantee the contractor will be able to submit VECPs that will support the Government's investment. This is likely to occur when a system is new and has a relatively unstable design and manufacturing process. However, recall from Figure 1 that the greatest potential for cost savings occurs in the earliest stages for design where the need for Value Engineering is the greatest. When an item or system has a relatively stable design or manufacturing process, the voluntary or incentive clause would be considered more appropriate. [Ref. 12: P.17]

The VECP is submitted to the contracting officer as a detailed justification which outlines and documents exactly how contract savings can be achieved. The VECP is the same thing as an engineering change proposal (ECP) with one exception. The VECP is specifically intended to produce cost savings for the contract while maintaining the original function of the item. It is a proposal that requires:

(1) A change in the instant contract against which the proposal is being submitted.

(2) Contract cost reduction without impairing desired functions provided that it does not involve a change:

(a) In deliverable quantities only.

(b) In research and development quantities or test quantities due solely to results of previous testing under the instant contract.

(c) To the contract type only.
[Ref. 29: P.3-3]

Table 1 lists the VECP share ratios. It can be seen that a contractor gains more when the voluntary approach applies.

Table 1
Government/Contractor Shares of VECP Savings
(All figures in percents)

	VE INCENTIVE (VOLUNTARY)		VE PROGRAM REQUIREMENT (MANDATORY)	
CONTRACT TYPE	INSTANT	FUTURE/ CONCURRENT	INSTANT	FUTURE/ CONCURRENT
FIXED-PRICE (other than incentive)	50/50	50/50	75/25	75/25
INCENTIVE (fixed price or cost)	*	50/50	*	75/25
COST REIMBURSEMENT (other than incentive)	75/25	75/25	85/15	85/15

* SAME AS THE SHARING RATIO IN THE CONTRACT FROM REF [11]

It should be noted that these ratios may be negotiable based on need and available funding.

When a contractor submits a VECP, the contracting officer has 45 days to accept it or reject it. If more than 45 days are required for the Government to review a VECP, the contracting officer is required to notify the contractor in writing detailing the reasons for the delay and the anticipated date a decision is expected to be made. The VECP may be accepted in complete or partial form. The decision to accept or reject a VECP or the determination of collateral cost or collateral sharing rates are not subject to the disputes clause. [Ref. 11: P.48-3]

When the VECP is submitted for review and approval, the following will be evaluated to determine the feasibility of the proposal:

(1) The relative merit of the proposed change versus the unchanged item.

(2) The technical competence of the personnel and the facilities required to accomplish the change.

(3) The manhour backlog to incorporate changes that have already been approved.

(4) The affect of spares, repair parts, data, and publications.

(5) The affect on the delivery schedule.

(6) The affect on training and training equipment.

(7) The affect on test and support equipment.

(8) The availability of funds.

(9) The affect on reliability and maintainability.

(10) The return on investment [Ref. 12: P.48]

If the VECP is approved, the contracting officer will negotiate the amount of cost savings with the contractor. To determine what is fair and reasonable for both the Government and the contractor, the extent of the change to the contract as a whole must be determined. In other words, the impact on each of the affected cost elements when they are compared to the original contract must be taken into account. Other things to consider would include the impact on concurrent and future contracts involving the same type of item. It is sometimes difficult to determine exactly where savings should be defined when looking forward throughout the service life of a certain product since user requirements change over time.

It is in the best interest of the Government to avoid paying large sums of money for savings if the estimated life span of an equipment or system is actually shorter than original estimates predicted. It is clearly a challenge to gauge the true measure of contract savings when implementing a VECP, particularly when the nature of the contract is extremely technical. Great care must be taken when determining the relative change a VECP has on the contract and the corresponding savings attributed to that change in order for the Government to realize maximum value from the Value Engineering clause.

For major weapon systems procurements, the FAR requires a Value Engineering Program Requirement Clause (VEPRC) for initial production buys. Figure 2 outlines the basic acquisition framework for the service life of a major weapon system. The acquisition process begins with the determination of a mission need and progresses through five distinct milestones and phases. The Defense Acquisition Board (DAB), chaired by the Under Secretary of Defense (Acquisition and Technology) USD(A&T), conducts an exhaustive milestone review to determine:

(1) Where the program is versus where the program should be;

(2) Where the program is going and how the Program Manager proposes to get there;

(3) What risks exist in the program and how the Program Manager will identify and close those risks;

(4) Is the Program Manager's proposed approach affordable. [Ref. 28: P.11-C-1]

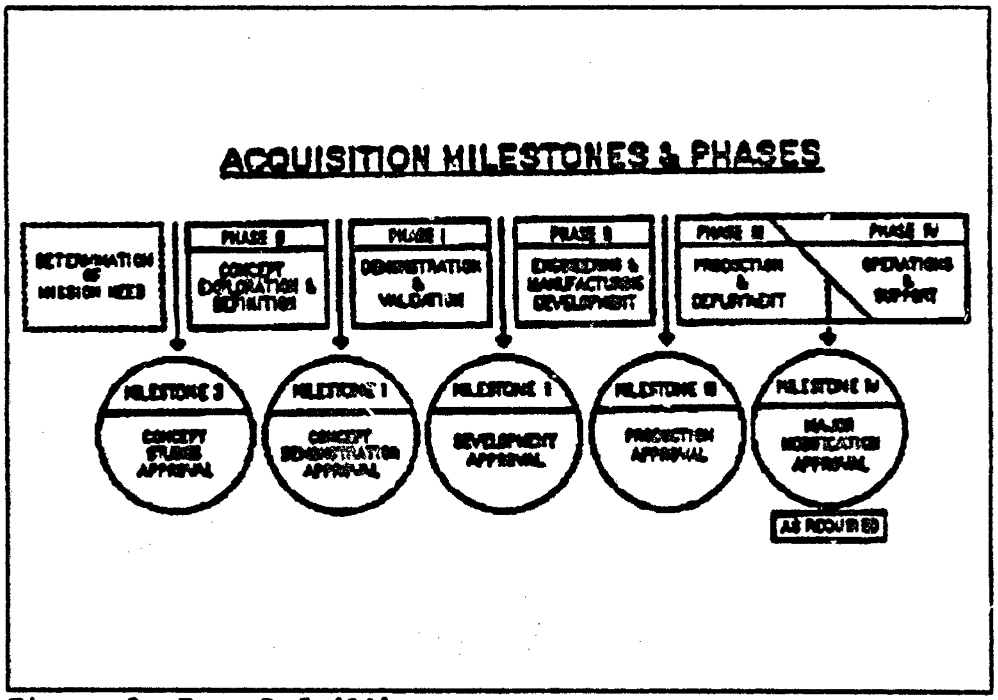

Figure 2 From Ref [28]

When the USD(A&T) is confident that all the pertinent issues have been addressed, he will grant approval for the program to proceed to the next phase. The introduction of VE is required at Milestone III, production approval. The VEPRC is not required when in the contracting officer's judgment, the contractor has demonstrated an effective Value Engineering program or the contract award was based on adequate competition. [Ref. 11: P.48-2]

2. Office of Management and Budget Circular No. A-131

To effectively carry out the Congressional requirements outlined in the FAR, the Executive branch communicates specific instructions to all Federal Agencies via OMB Circulars. On 21 May 1993, the then Director of the Office of Management and Budget, Leon Panetta, released the latest update regarding Value Engineering which requires additional procedural emphasis in reporting and recordskeeping, planning and review, and funding considerations in annual budget requests to OMB. Additionally, the use of Value Engineering is now required to include the use of a product, service, and process improvement orientation. [Ref. 35: P.2] The Value Engineering emphasis which focuses on a process improvement orientation is more conducive to a software development environment. Specifically, Circular A-131 requires Federal Agencies to:

> Use Value Engineering as a management tool, where appropriate, to ensure realistic budgets, identify and remove nonessential capital and operating costs, and improve and maintain optimum quality of program and acquisition functions. Senior management will establish and maintain VE programs, procedures, and processes to provide for the aggressive, systematic development and maintenance of the most effective, efficient, and economical and environmentally-sound arrangements for conducting the work of agencies, and to provide a sound basis for identifying and reporting accomplishments. [Ref. 35: P.2]

OMB Circular A-131 encourages the use of other management techniques in conjunction with Value Engineering to achieve reduced costs. These techniques include, but are not limited to, design-to-cost, total quality management, life cycle costing, and concurrent engineering. [Ref. 35]

3. Value Engineering Guidance For DOD and Navy Use

With OMB A-131 directing Federal agencies to use Value Engineering, DOD implements its Value Engineering Program through DOD Instruction 5000.2 of 23 February 1991 which is policy and procedure for acquisition management. DOD policy is to require Value Engineering in the design for manufacturing and production. Reporting and format requirements are also listed to enable the DOD components to submit their annual statistical summaries of Value Engineering accomplishments to OMB. [Ref. 28]

Major systems commands within the Navy draft Value Engineering instructions which are tailored to their individual organizations. For example, the Naval Air Systems Command (NAVAIR) promulgates its policy in an instruction to all headquarters and field components of the Naval Aviation Systems Team (TEAM). This instruction incorporates guidance directly from the FAR, OMB Circular A-131, and DOD Instruction 5000.2. Specific VE responsibilities are outlined for senior management personnel and individual field activities within NAVAIR in order to implement an effective Value Engineering program. [Ref. 30]

E. VALUE ENGINEERING: A NEW DIRECTION

1. The Perry Memorandum

In June 1994, the Secretary of Defense, Dr. William J. Perry released a memorandum which directed a new way of doing business in DOD with regards to specifications and standards. This memorandum directs the use of performance specifications for programs in any acquisition category. When performance specifications cannot meet requirements, then non-Government standards or commercial standards are required. In the event performance or commercial standards cannot be used to satisfy an acquisition requirement, a waiver must be granted by the Milestone Decision Authority for use of a military standard or specification. Replenishing existing inventories do not require waivers. [Ref. 22]

Dr. Perry also encourages the use of the Value Engineering no-cost settlement method in existing contracts. The FAR discusses the no-cost settlement method as follows:

> To minimize the administrative cost for both parties where there is a known continuing requirement for the unit, consideration should be given to the settlement of a VECP submitted against the VE incentive clause of the contract at no cost to either party. Under this method of settlement, the contractor would keep all of the savings on the instant contract, and all savings on its concurrent contracts only. The Government would keep all savings resulting from concurrent contracts placed on other sources, savings from all future contracts, and all collateral savings. Use of this method must be by mutual agreement of both parties for individual VECPs. [Ref. 11: P.48-4]

The significance of using performance and commercial specifications is significant to Value Engineering. In a Value Engineering study, all aspects of the item or process are challenged to suggest alternatives that would either lower cost, increase value, and maintain function. By eliminating military specifications and standards, the ability to suggest alternatives is expected to be considerably less restrictive. This is because performance and commercial specifications have the potential to offer a wider range of alternatives to achieve a higher degree of value for an item.

2. 104th Congress H.R. 719

On 27 January 1995, Representative Collins of Illinois introduced H.R. 719 which is cited as the "Systematic Application of Value Engineering Act of 1995." This legislation, if enacted, will require the development of criteria to assist Federal and contractor employees in identifying projects that have the highest potential for savings when the methodologies of Value Engineering are applied: H.R. 719 emphasizes the need to apply Value Engineering in the early stages of development or design of an item or process in order to reduce life-cycle costs. Two rather enterprising proposals in this bill regarding Value Engineering acquisition savings are:

> (A) Fifty percent shall be available to the agency for project, system, or development; and use for programs in effect on the date of the enactment of the Act under which incentives are provided to employees of the agency to identify and implement methods for achieving savings in programs, projects, systems, and product development of the agency.
>
> (B) Fifty percent shall be deposited in the general fund of the Treasury and used to reduce the Federal debt. [Ref. 36]

When used appropriately, Value Engineering is recognized as an effective cost saving tool. The legislative language in H.R. 719 indicates that Congress fully supports the concepts of Value Engineering and intends to ensure that it receives appropriate management attention throughout the Federal Government.

F. A VALUE ENGINEERING SYSTEMS EXAMPLE

At this point it is appropriate to provide an example of the application of Value Engineering to a system or process. This is done in order to assist the reader in relating the basic concepts of Value Engineering to a software development process.

Purchasing agents for the State of New Mexico were tasked to reduce the amount of mailing costs for standard documents that were regularly mailed to their state residents. A Value Engineering study was conducted to analyze the complete function of the entire mailing process. Key personnel such as systems analyst, buyer, and office personnel were invited to participate in the study so that inefficient costs could be challenged. The resulting changes to the old system saved the state in excess $250,000 per year. Significant changes included:

> (1) Redesigning and reducing the number of forms used.
>
> (2) Producing standard forms in-house vice purchasing them from commercial sources.
>
> (3) Programming computer operated mailing systems to mail multiple documents in one envelop to the same address vice mailing single documents multiple times. [Ref. 8: P.575]

Regardless of the system or process that is involved, none are perfect and inefficiencies or alternatives can always be challenged in order to increase value to the end user. The same holds true in theory to a software development process. Software process improvement is a continual assessment of development practices which seeks to eliminate inefficiencies and introduce refinements. It is difficult to define and institute an organization wide process that complies with the goals of Software Engineering Institute's

Capability Maturity Model (CMM). Each software development project is driven by very intelligent human beings, but by their very nature they have yet to achieve levels of "ultimate perfection" beyond that defined in the CMM. [Ref. 20]

G. SUMMARY

Value Engineering is a relatively easy concept to understand. It is well documented throughout Federal Government and when used appropriately it is a proven method to reduce cost and increase value to the end user. To be successful Value Engineering requires constant management attention to achieve acceptable levels participation. Despite the high levels of success Value Engineering has experienced in the past, additional emphasis is warranted to deal with the financial realities associated with the post Cold War era.

Chapter III will discuss Value Engineering concepts that involve software development and acquisition.

III. SOFTWARE VALUE ENGINEERING APPLICATIONS

A. OVERVIEW

As stated earlier, the rising cost of software acquisition is consuming an increasing percentage of the DOD budget each year. In 1994, DOD software costs were approximately 42 billion dollars and future costs will continue to escalate. [Ref. 20] As weapon systems become more complex, the software development effort required to successfully field these systems increases. It is not uncommon for the development of software in today's weapon systems to experience cost overruns, schedule delays, and performance problems. These problems can be the result of inadequate management attention, ill-defined system requirement, and inadequate testing. With the high cost of such weapon systems, reasonable expectations from the public demand fully functional systems. [Ref. 32: P.1-3]

Value Engineering is recognized as an effective cost cutting technique for weapons and systems programs, but has not been used to its fullest extent. [Ref. 34: P.2-4] The National Performance Review has recommended using performance based contracting incentives such as Value Engineering bonuses to encourage better vendor performance for Information Technology procurements.

Value Engineering and Software Engineering are both "people" businesses in that both disciplines require exceptional "thought processes" in order to be successful. However, there is little evidence to suggest that the methodologies of Value Engineering actually support the acquisition and development of software within DOD. Software related VECPs are extremely rare because it is not widely accepted that Value Engineering applies to software. [Ref. 21: P.270]

This chapter will present a basic application of the methodologies of Value Engineering to software acquisition. Further analysis will cover actual applications of software Value Engineering.

B. VALUE ENGINEERING CONCEPTS

To consider the application of Value Engineering to software, a review of the basic definition is required. The FAR defines Value Engineering as,

> ...an organized effort to analyze the functions of systems, equipment, facilities, services, and supplies for the purpose of achieving the essential functions at the lowest life cycle cost consistent with required performance, reliability, quality and safety. [Ref. 11: P.48-2]

This fairly straightforward definition has a very basic meaning that does not appear to communicate any limitations to the reader. This definition fulfills the explicit purpose of Value Engineering as presented by Mr Miles by identifying unnecessary cost in an item and making a decision to eliminate that cost. [Ref. 19: P.viii]

At this point it is appropriate to introduce the definitions of software and software engineering to this analysis. Software is defined by Webster's dictionary as

> 1. Written or printed data, as programs, routines, and symbolic languages, requisite to computer operations. 2. Documents containing information on computer operations and maintenance. [Ref. 27: P.1105]

Software engineering is defined by Barry W. Boehm in Software Engineering Economics as:

> ...the application of science and mathematics by which the capabilities of computer equipment are

made useful to man via computer programs, procedures, and associated documentation. [Ref. 4: P.16]

Mr Boehm amplifies this definition one step further by stating that a good software engineer produces software that is "useful to man." To be "useful to man" means that software is affordable and performs functions required by society. The central theme of his book appears to revolve around the software engineering concept of controlling software cost in order to fulfill human needs. [Ref. 4: P.17]

1. Software Development

Each software development project begins with a review of available techniques and processes that are used to develop software. To ensure quality software is produced, appropriate management attention must be directed in the areas with the potential to generate challenges beyond original expectations. Software development is extremely complex and can be compared to the construction of aircraft. Aircraft manufacturers use a variety of tools, materials, and processes to develop an aircraft consistent with contract requirements. Both software and aeronautical engineers constantly look for new techniques and processes to improve their products. [Ref. 2: P.1]

Figure 3 outlines the steps typical of software development for a large scale software project. Each step contributes to the production of specific software products. [Ref. 3: P.2-2]. These products are usually associated with a list of functional requirements that evolve from incomplete drafts to highly detailed specifications [Ref 13: P.24].

It can be seen from Figure 3 that software development can be a complicated process. In order to apply the methodologies of Value Engineering, specific software engineering methodologies must also be reviewed so that similarities can be analyzed.

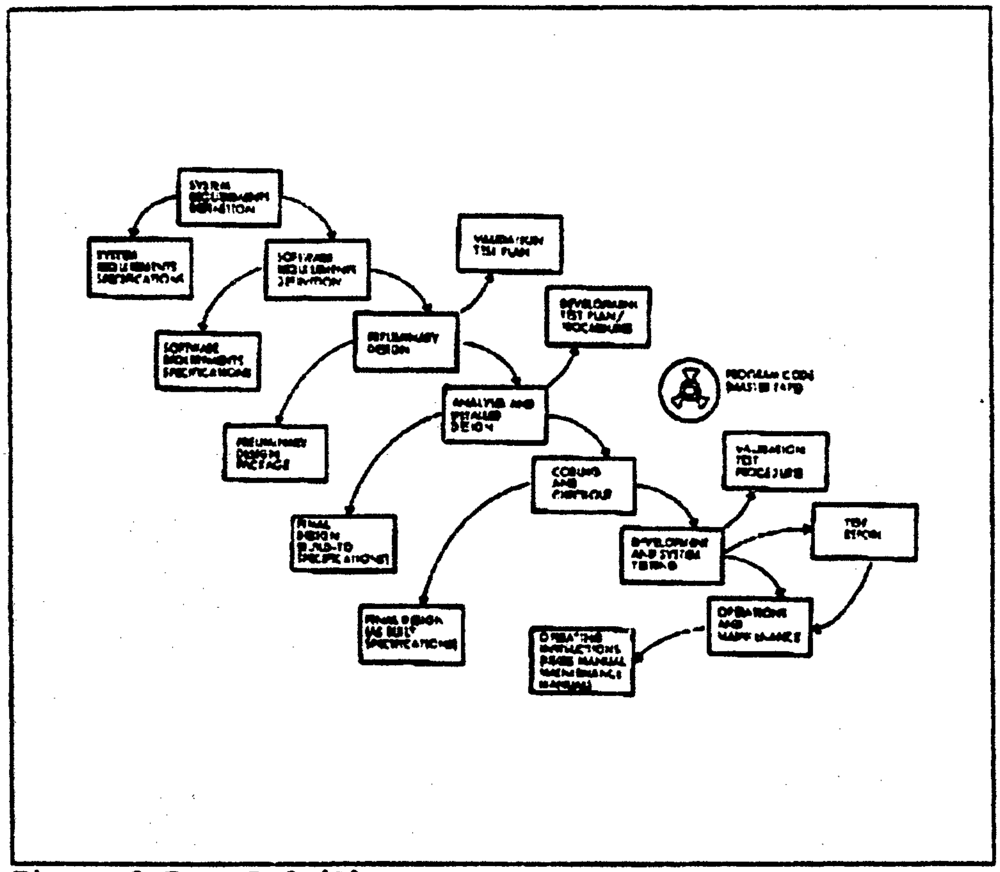

Figure 3 From Ref [3]

The sections that follow will compare software engineering methodologies with applicable Value Engineering methodologies.

a. *Weinberg's Experiment*

In software engineering, an honorable goal for a software developer would be to provide software products that fulfill the needs of a customer in a uniform manner. In other words, if a customer specifies that he desires a software product to exhibit certain characteristics, such as program efficiency or program clarity, then the customer would normally not expect other characteristics to suffer as a result of preferring one over others. This software engineering concept is known as "The Plurality of Goals." The plurality of goals means that different software goals conflict with each other in software development. This means that if one particular software goal receives special emphasis, then other software goals will most likely suffer as a result. [Ref. 4: P.20-21]

In Wienberg's experiment, five teams were given a programming assignment. The assignment was the same for each team, however, each team was to place special emphasis on a specific software goal. The teams were each given a different a goal to concentrate on. The results showed that all the other software goals analyzed suffered as a result of concentrating on one specific goal. Figure 4 shows the results of the experiment in which the plurality of goals is demonstrated. Mr. Boehm points out that the first team whose objective "effort to complete" shows that

> ...pure concentration on minimizing the software development budget and schedule is likely to have bad effects for software life-cycle budget. [Ref. 4: P.21]

From a Value Engineering standpoint, one would question and challenge every aspect of the plurality of goals in order to achieve the highest levels of value. One applicable Value Engineering technique is to "Use information from only the best source." Some useful questions that should be asked would include:

- Why is memory our most important software goal?

- How important is program clarity to our organization?
- Why should output clarity be considered?
- Would other software goals better suit our needs and why?
- Who can provide the best answers; the software engineer or the customer?
- What are our real requirements, will something else suffice? [Ref. 19: P.48]

Team objective: Optimize	Resulting Rank on Performance[a]				
	Effort to Complete	Number of Statements	Memory Required	Program Clarity	Output Clarity
Effort to complete	1	4	4	5	3
Number of statements	2-3	1	2	3	5
Memory required	5	2	1	4	4
Program clarity	4	3	3	2	2
Output clarity	2-3	5	5	1	1

Figure 4 From Ref [4]

This type of value analysis may result in a positive value change to the original requirement. Weinberg's experiment demonstrates that all software characteristics are not represented uniformly in the development process as one would think. This makes it essential that the user is provided the opportunity to examine their requirements in detail. In software development, user involvement requires additional emphasis due to inaccurate or incomplete requirements definition, conflicting needs, user resistance, and communication breakdowns [Ref. 20: P.32]. Thoroughly challenging all relevant aspects regarding the software goals, as suggested above, is required to provide useful information that will lead to decisions that will result in additional value to the customer. [Ref. 19: P.48]

b. Marginal Analysis

One method to assess value in software engineering is to graph value relative to cost. Mr. Boehm discusses this approach by stating,

> The total value (TV) of an information processing system is its effectiveness when expressed in the same units used to express the cost (C) of the system. In that case, the net value (NV) is defined as the effectiveness-cost difference, NV=TV - C [Ref. 4: P.208] [Parentheses added]

Figure 5 shows the cost function C(x) graphed with the Total Value function TV(x). It can be seen that for any given activity, Net Value is maximized at activity level X_{max}, which is where Net Value has the largest positive value. [Ref. 4: P.206]

Figure 6 shows the net value function which could be derived from Figure 5 simply by subtracting total value from cost. Notice how when NV is negative, the activity is undergoing a phase of investment. When NV is positive, the activity level is profitable. Beyond activity level X_4, the organization is no longer profitable and has over invested.

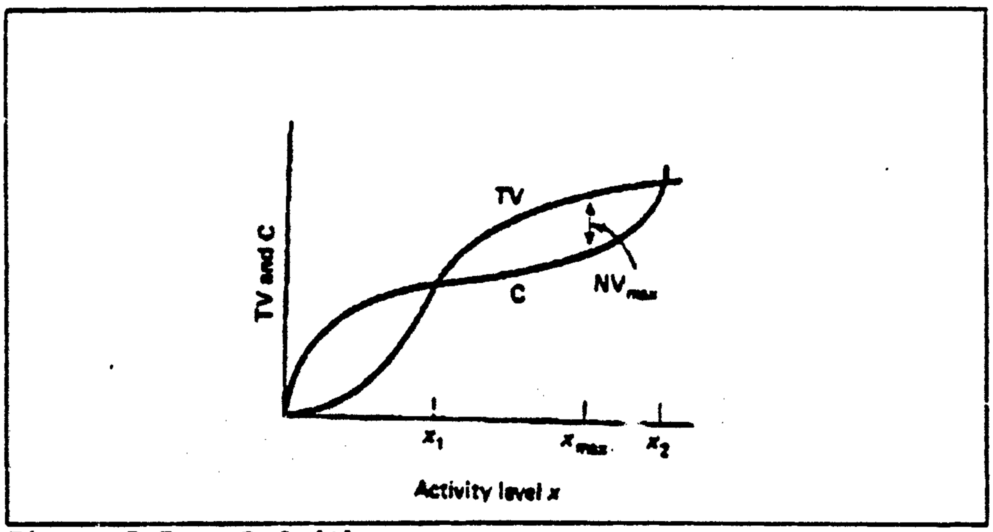

Figure 5 From Ref [4]

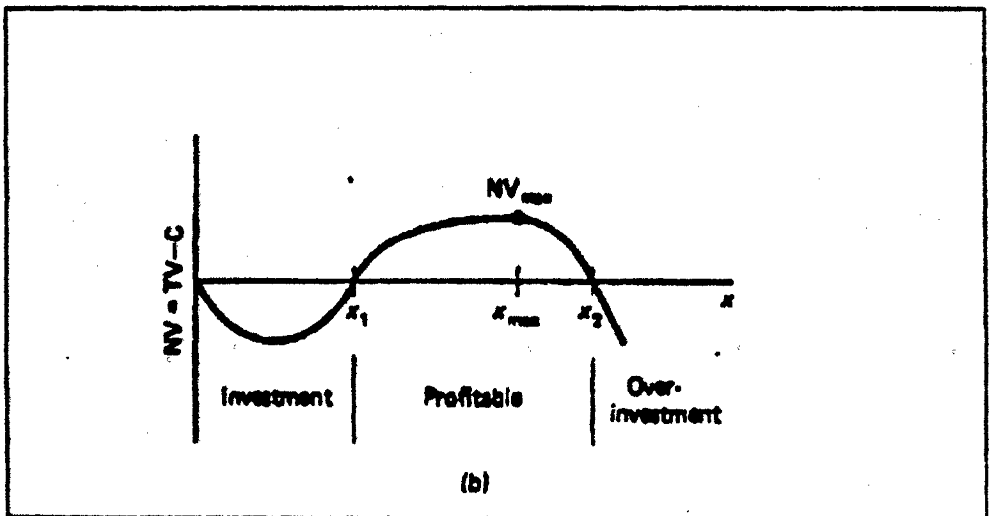

Figure 6 From Ref [4]

Mr Boehm points out that the slope of the NV curve or marginal net value (MNV) can be used for decision making purposes. According to Boehm, three conclusions can be made from the MNV:

> 1. If the MNV is positive, increase the activity level.
>
> 2. If the MNV is negative, decrease the activity level.
>
> 3. If the MNV is zero, the activity level is optimal (X_{max}). [Ref. 4: P.209]

In a Value Engineering study it is beneficial to "Get all available cost". For cost to be meaningful, accurate figures must be developed to make informed management decisions. Cost behavior will fluctuate depending on the nature of an organization's cost elements and the activity level at which it operates. Rates such as overhead, labor, and material, can exhibit variations at different levels of output. Rates can also impact cost as evidenced by various levels of efficiency or economies of scale. As activity levels change, each change in cost should be questioned to determine its true meaning and its resulting impact upon the organization. Mr Miles states the importance of getting all available costs:

> Without meaningful cost, decisions will not, and cannot, be made to provide good value.
> [Ref. 19: P.45]

To accurately collect the cost data presented in Figures 5 and 6, a thorough analysis of all relevant cost data must be reviewed for all activity levels as discussed earlier in order for these graphical depictions to be meaningful and accurate for the user. This is true for all organizations since cost structures for each software development project in an organization is completely different from any other software development project. [Ref. 19: P.42]

Value measurements cannot be extracted from an organization's accounting ledgers, but must be determined from an assessment of utility or worth. Recall that value is a measure of utility and worth resulting from some detailed analysis which satisfies a recognized need. [Ref. 18: P.23]

c. *Goldplating*

Any product, whether it is hardware or software can experience excessive cost in development simply by adding "nice to have items" that do not add value to the customer. In software engineering, "goldplating" makes the job larger and adds costs which are disproportionate to original software requirements. One common method to make the software job larger, and increase the cost significantly, is to succumb to the temptation to add additional software engineering to a large software project. [Ref. 4: P.191]

According to Boehm, "goldplating" can result from adding unneeded features to software requirements. Three of his examples include:

> (1) Instant Response Time: Overloading processing systems with rapid response times for all transactions that exceed user requirements.
>
> (2) Pinpoint accuracy: Requiring systems to produce mathematical calculations to 4 digit accuracy versus 2 digit accuracy.
>
> (3) Everything for everybody: Systems developed which provide a corporation's entire information processing needs into one comprehensive integrated system. [Ref. 4: P.192]

In Value Engineering, "Using industry specialists to extend specialized knowledge," could be used to challenge the tendency to "goldplate" software products [Ref. 19: P.71]. Getting appropriate levels of value in a product or system requires challenging all known alternatives in order for managers to make value decisions. Appropriate questions to ask in this category should include:

- What is the exact function the software must perform and why?
- Who is in the best position to define the unique requirements of the software and why?
- What software costs are associated with identified functional requirements and why?
- What other software solutions will satisfy identified requirements and what are their relative costs? [Ref. 19: P.71]

Asking pertinent questions that are exhaustive in nature will challenge all relevant people that interact with an organization to specify minimum unique requirements. This will ensure that appropriate levels of value will be achieved.

For example, an airline reservation processing system must have an adequate response time capable of processing a predetermined or historical number of transactions per day. Value would be lost if the processing system had the capacity to process double or triple the maximum amount of transactions that are incurred on a day-to-day basis. On the other hand, a reservation clerk has to have the right amount of time to process a transaction. What person is the best person to determine transaction processing time? The reservation clerk certainly is a key player because any clerk is physically limited in what one person can do in one day. Therefore, they must be able to process a minimum amount of transactions to be efficient. The customer or passenger waiting in a line also has some tolerance for the average amount of time they are willing to spend waiting in a line, or on the telephone, in order to service their requirements. The airline president is concerned that too much time is spent processing transactions and not enough time filling available seats on all flights. Flight attendants must know how much food and beverages to stock for each flight in order to meet customer demand for those who book reservations at the last minute, yet control cost goals at the same time. Operational analysis personnel need flight data to study trends to ensure resources are properly allocated to the most profitable routes traveled. The Federal Aviation Administration requires airlines to produce passenger lists immediately after takeoff in the event of an emergency. [Ref. 5: P.142]

These examples show how many different types of people can be involved when considering minimum software requirements for an information processing system. To ensure optimum levels of value, the influence of a system should be analyzed against all applicable elements of an organization. This will provide management useful information that will indicate to what extent their requirements have been met and what may need to be done to eliminate unnecessary requirements and inefficient cost. [Ref. 19: P.71]

d. *Legacy Systems*

One thing all businesses inevitably experience over time is change. To sufficiently meet the needs of a competitive business environment, software systems must also be capable of changing. However, those software systems that do not keep up with the changing times are classified as legacy systems. Legacy systems are informally defined as, Large software systems that we don't know how to cope with but that are vital to our organization. [Ref. 1: P.19]

Replacing legacy systems in a business organization requires a thorough analysis of how change has evolved over time. Software contains the rules of an organization that have accumulated over time, so a review of the software system rather than the human processes involved is required. In an organization normal business operations change over time, however, efficient change is impeded if software systems are not updated. Organizations need to view software evolution as an integral part of software development. [Ref. 1: P.19-21]

Avery Division of Avery Denninson, a $2.6 billion office supplies firm in Pasadena, California uses a company developed technique known as Avery Value Analysis as a solution to their legacy systems. Their approach is to consider all of the opportunities for changes in business processes, cost of processes, and alternative processes. Cross-functional teams use brainstorming techniques as a means to compare and evaluate differences in their legacy and manual systems. This allows Avery to identify where changes, such as outsourcing requirements, need to be made in order to obtain better value. [Ref. 10: P.88]

Another method for an organization to apply Value Engineering to legacy systems, or any other software system, is to analyze the business value of the system. Table 2 represents a system broken down by key business applications and objectives. The rows represent business applications and the columns represent business objectives. This representation allows business applications to be compared in relation to business objectives. A weighted score is assigned to each business application based on relative importance or value to the organization. Complete rankings can then be prioritized for business applications in a manner that translates numerical rankings into value rankings that management personnel can use as a decision making tool. [Ref. 25: P.28]

This method of Value Analysis provides management with the best information possible in a manner that is readily interpreted. Remember the Value Engineering technique that suggest to use only information from the best source. Cross-functional teams provide the best information possible because these teams are individuals who are most qualified or closest to the problem. People who work directly with the relevant aspects being analyzed are better suited to provide more reliable information to management personnel. Management must take this information and consider this input in terms such as reliability, credibility, and risk in order to make value decisions. [Ref. 19: P.49]

Table 2

BUSINESS VALUE ANALYSIS				
Application	Market Value	Profit Contribution	Information Significance	Score
Credit/Debit	10	10	10	30
Sales Support	40	30	20	90
Inventory	10	30	20	60
Accounting	10	10	30	50
Order Entry	30	20	20	70
Total	100	100	100	300

Source: Ref [25]

Using a scoring system, such as the one presented above, to assess value is an effective tool with unlimited applications. For example, the U.S. Air Force implemented a [illegible] system technique to highlight inspection procedures which determined the need for maintenance requirements of all C-130A aircraft wings. These maintenance requirements were intended to correct structural deficiencies such as corrosion, cracks, and previous structural repairs. Each structural deficiency was assigned a point value determined by the severity of the defect. C-130A aircraft wings with a score of 3000 or more required a complete wing overhaul. The success of this Value Engineering technique permitted the U.S. Air Force to drop the C-130A wing rehabilitation program from its annual budget. Total estimated savings over a three year period were calculated at over $68 million. [Ref. 15: P.11]

e. Software Reuse

To consider software reuse as applicable to Value Engineering, a review of the FAR is appropriate to define useful techniques. The FAR states that Value Engineering is,

> The formal technique by which contractors may (1) voluntarily suggest methods for performing more economically and share in any resulting savings or (2) be required to establish a program to identify and submit to the Government methods for performing more economically. [Ref. 11: P.48-2]

Software reuse is the practice of using existing software assets to develop new applications. Reusable software can be code segments, specifications, design, test data and test plans, software tools, or anything associated with software. Software reuse can be viewed as a method to reduce software development cost while improving software quality and reliability. [Ref. 33: P.2-5]

The concept of reuse is common to all engineering disciplines. Products manufactured are usually developed using [illegible] techniques, designs, and processes which have been previously brought into existence. For example, aircraft and automobiles are not manufactured from scratch. Existing designs, techniques, and knowledge are commonly reused in the manufacturing process to facilitate product development. In a similar manner, manufacturing reuse can be adapted to software development, maintenance, and acquisition processes. [Ref. 26: P.1]

Congress and DOD have repeatedly stressed the importance of reducing software development and acquisition cost. Software reuse has proven to be a cost effective tool for both DOD and industry. The Reuse Executive Primer produced by the Software Reuse Initiative Program Management Office in February 1995 noted the following software reuse achievements:

> The Navy experienced a 26% reduction in required labor hours to develop and maintain its Restructured Naval Tactical Data Systems (RNTDS).
>
> Raytheon saw a 50% increase in productivity in its Missile Systems Division.
>
> Fujitsu's Software Development for Electronic Switching Systems (ESS) began delivering 70% of its ESSs on schedule (as opposed to only 20% before adopting reuse principles).
>
> The Army estimates a cost avoidance of $479.9 million for its Tactical Command and Control System, allowing additional mission requirements to be addressed during a period of funding short falls. [Ref. 26: P.2]

Value Engineering can be applied to software reuse by encouraging a contractor to utilize reuse to the maximum extent possible. Normally, a contractor is paid for the effort they spend writing the software. As additional lines of code are written, additional earnings are produced. To reuse code, a contractor can retrieve existing software from the DOD Reusable Ada Packages for Information systems

Development Center Library (RAPID). When reuse is substituted in the software development process, time and effort can be reduced substantially. Value Engineering results by translating the resulting time and effort into acquisition savings which can be shared between the Government and the contractor. Although this usually means less money paid to a contractor as a result of contract cost reductions, profits can still be significant through cost elimination. [Ref. 14: P.38]

Although Value Engineering works well in theory for software reuse, it is difficult to apply in practice. The new military standard for software development and documentation is MIL-STD-498 which superseded MIL-STD-2167A. The purpose of MIL-STD-498 is to establish uniform requirements for software development and documentation. MIL-STD-498 requires contractors to identify and evaluate reusable software products for potential use when competing for a contract. This information is documented in the contractor's software development plan. Obviously, this prevents a contractor from voluntarily suggesting an economical software reuse alternative in the name of Value Engineering until after the contract has been awarded. However, depending upon the competition involved in a particular contract, the Government has the advantage of being more informed than the contractor about the potential number of software reuse opportunities because all the offerors competing for the contract identify known software reuse products applicable to the requirement. [Ref. 6: P.8]

f. Software Maintenance

Software maintenance represents a significant portion of software life-cycle costs. The cost associated with maintaining software in DOD after fielding is approaching

70% of the total software cost. A critical management concern in DOD is that the requirements for software maintenance personnel is estimated to increase by 12% per year. However, the current availability of qualified personnel to satisfy software maintenance requirements is only increasing by 4% per year. [Ref. 9: P.7-2]

Maintaining software is not the same as maintaining hardware. Hardware requires maintenance because components eventually break down over time. When hardware components are replaced, the item's original configuration remains intact. When software requires maintenance, a new software configuration is created because code must be modified or added to support a required capability. [Ref. 9: P.7-5]

At this point it is appropriate to define software maintenance. Software maintenance is defined as,

> The process of modifying existing operational software while leaving its primary functions intact. Software maintenance is classified into two main categories: (1) Software update; which results in a changed functional specification for the software product and (2) Software repair; which leaves the functional specification intact. [Ref. 4: P.534]

For cost and planning considerations, software maintenance considerations are critical during the planning and requirements phase of development. If an error or desired software change is detected early in the development phase, the problem is relatively easy to correct. When errors are not detected until after the software is fielded, the correction and effort required is very significant. Changes must be made to records such as maintenance, training, and operational manuals. Additionally, there is an administrative burden that is also incurred since management attention must usually address any changes to organizational matters in a formal manner. Corrections that are not made in the planning and requirements phase can cost up to 100 times more to fix after the software has been fielded. [Ref. 4: P.40]

(1) Design Analysis/Value Analysis Checklist. In Value Engineering, two conceptual tools that are useful in determining value are design analysis and value analysis checklists. Design analysis is a methodical step-by-step design review of an item which is then compared to the function the item performs. A design analysis determines whether an item can be eliminated, simplified, substituted, or changed to facilitate a more economical production process. A value analysis checklist analyzes the function of an item against an extensive checklist which is used to evaluate the item's value. Any question on the checklist which does not provide a satisfactory answer regarding an item's value is challenged until an acceptable alternative can be found that provides satisfactory improvement. [Ref. 4: P.562-563]

In software development, future maintenance actions should be considered as early as possible to reduce total software life cycle cost. Any product, whether it is software or hardware, can be designed to facilitate future maintenance. Two positive examples which could aid future maintenance for a software product include:

> A document in which pages, figures, and tables are numbered by major headings, e.g. 1-1, 1-2,...,2-1, 2-2,..., so that insertions and deletions may be made without renumbering the entire document. [Ref. 3: P.3-10]

> A program which is deliberately designed to fit into less than the available resources (core, disk, tape, mass storage, etc.), so as to leave room for modification. [Ref. 3: P.3-10]

A design analysis focuses on an item's function and cost. In software engineering, the design phase of software development is the production process. This is conducive to software design analysis since its documentation must exhibit functional completeness and traceability back to functional specifications. To control future maintenance cost, software maintenance should be easily identified and controlled to the maximum extent possible. [Ref. 2: P.164]

Future value of a software system can be preserved with a rigid design analysis made up of an extensive checklist of questions. This checklist can be developed using a Value Engineering technique known as brainstorming. Although the challenges to a design could be numerous, some questions a software maintainer might consider evaluating include the following:

> What is the overall design philosophy?
>
> Are the system's components coupled as loosely as possible?
>
> Have parameters of the system in areas of likely future change been identified?
>
> Have existing re-usable components been identified for incorporation into the system?
>
> Are the future system maintainers satisfied with the extent to which the system design satisfies the maintenance design goals set for it?
>
> Are the future system maintainers fully familiarized with the design philosophy? [Ref. 2: P.164]

A value analysis checklist can also be generated using brainstorming techniques which can focus on potential maintenance areas that impact future value. Again, questions that could be evaluated to analyze future value include:

> What are the performance capabilities of the system and how might they grow?
>
> Which areas of requirement might become unnecessary and removable?

> Is there the possibility of a need to have different versions of the system with different capabilities?
>
> What are the implications on manpower and computer resources if the identified changes come about?
>
> Which areas of function have the greater chance of requiring change in light of experience with the system. [Ref. 2: P.104]

(2) US Army Software Value Engineering Study. In 1987, the U.S. Army Communications Electronics Command (CECOM) initiated a Software Value Engineering study in order to implement master plans for each Army Command. The idea that Value Engineering could be applied to software engineering was based on the premise that the two disciplines were primarily people oriented. It was also thought feasible because software products might contain unnecessary or inefficient cost during software development and support processes. The study was:

> A joint effort between Value Engineers and Software Engineers to determine if and how the methodology of Function Analysis, the heart of the Value Engineering/Analysis discipline, can be applied fruitfully to software development and support. [Ref. 21: P.269]

The study found that fielded software products may include code that is inefficient, redundant, or dead. The cost of such code is clearly inefficient and unnecessary and some analysis is required to determine what, if anything, should be done to correct the code. In such a situation, relevant Value Engineering/Software Engineering questions should challenge the status of the code itself. The members of the study used a software orientated Value Engineering Job Plan that employed Functional Analysis System Technique (FAST) diagrams to document a common understanding of the software development and support processes.

In Barry Boehm's book on Software Engineering Economics, he describes the considerations and decisions involved when software maintenance is weighed against financial constraints. This is the basic thrust of the 1987 CECOM study. Essentially, management must decide how much maintenance activity a software product must undergo before its value deteriorates [Ref. 4: P.545]. To determine appropriate levels of value for an organization, a cost benefit or point of diminishing returns analysis might ask the following questions:

> At what investment are inputs being consumed without a great deal of resulting output?
>
> At what point of diminishing returns will additional inputs produce relatively little increase in output? [Ref. 4: P.189]

A significant finding in the 1987 CECOM study focused on the issue of human communication. Software projects are normally made up of more than one person. Different people on software projects, all of whom exhibit fallible human traits, interact together to pursue a common goal. As the number of people in a project becomes larger, communications become more difficult and result in diseconomies of scale. Figure 7 shows how increasing a project can contribute to problems in communications.

Boehm points out that in a software project with N people, there are $N(N-1)/2$ potential interpersonal communication paths. For example, with $N=4$ people there are 6 different communication paths possible. With more people on a project there are more possibilities for social differences to disrupt communication and impede efficiency. The only way to avoid these problems is to reduce the number of people on the project as much as possible. [Ref. 4: P.190]

Involvement from maintenance personnel early in the system definition stage of software development was found to effectively reduce total software life cycle cost. This finding alone is the corner stone of Value Engineering.

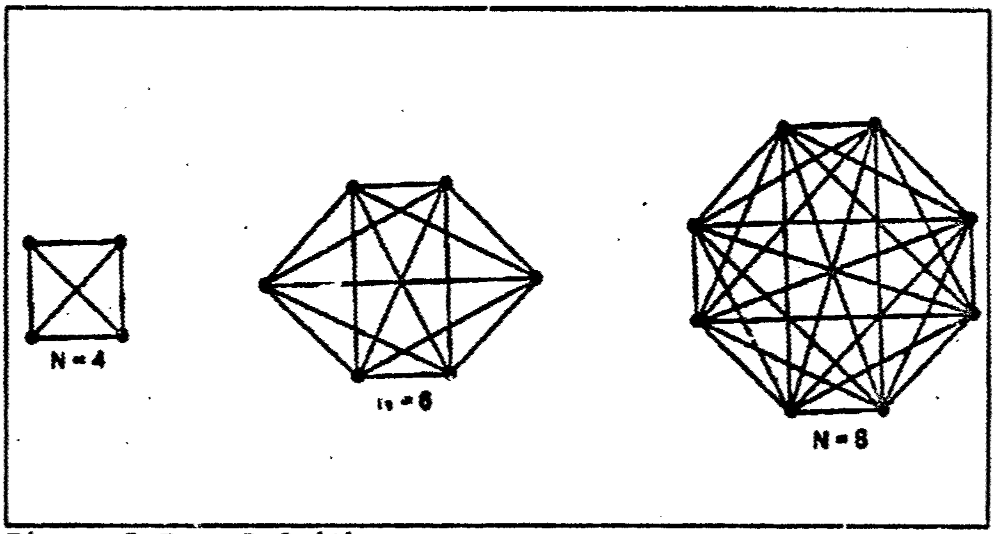

Figure 7 From Ref [4]

Recall from Figure 1 that Value Engineering is best applied in the R&D and design stages of development because management decisions will "lock cost" into the item. This is a crucial lesson particularly with the computer age experiencing rapid increases in the development of software languages, processes, and products. [Ref. 4]

C. SUMMARY

Throughout this chapter the methodologies of Value Engineering were applied to the development of computer software. The approaches taken for each of the examples cited above demonstrates what Value Engineering is and what it is not. It is important to remember that Value Engineering is a valuable tool because it is a philosophy and not a science and therefore makes it applicable to such processes as software development and acquisition.

Chapter IV will present and analyze the perceptions of DOD program managers and software engineers to determine the extent to which Value Engineering can be successfully applied to software development.

IV. DOD PROGRAM MANAGER AND SOFTWARE ENGINEERING PERCEPTIONS OF VALUE ENGINEERING IN SOFTWARE DEVELOPMENT

A. INTRODUCTION

This chapter will present and analyze the perceptions of DOD program managers and software engineers with regard to the application of Value Engineering methodologies to software development. The information in this chapter was primarily accumulated through the use of 335 surveys mailed to DOD Program Managers, U.S. Navy Systems Commands, and Defense Contract Management Command Districts. There were 81 responses which represent a return rate of 24%.

The questions used in the surveys were designed to provided a brief description of how Value Engineering might be applied to software. The survey was deliberately designed to include six questions as requested by various program managers queried during initial research investigation. A copy of this survey can be found in the Appendix. Responses were primarily written by software engineers or personnel with extensive software experience.

All responses provided some information that was unique in some way to the respondents' personal opinions or to the working environment of the military commands to which they were assigned.

B. SURVEY QUESTIONS AND ANALYSIS

1. Question one

The Federal Acquisition Regulation (FAR), Part 48, discusses Value Engineering (VE) requirements. In your opinion, does VE apply to computer software? If your answer is no, please state specifically what subpart of FAR Part 48 precludes the use of VE in computer software acquisition and why. If your answer is yes, please state specifically what subpart of FAR Part 48 does apply to software acquisition and why it does.

Response: Among the responses, 71% indicated that FAR Part 48 does apply to computer software, 15% indicated that FAR Part 48 did not apply, and 14% provided no response.

All the surveys that reported Value Engineering does apply to software acquisition effectively communicated strong support for the VE/software relationship in the FAR. Those that responded favorably provided one or more of the following remarks (frequency of remark is indicated in parenthesis):

1. FAR Part 48 can be interpreted to apply to software based on the concept or general definition provided in the FAR: "VE is the formal technique by which contractors may voluntarily suggest methods for performing more economically and share in any resulting savings." (15)

2. There is nothing in FAR Part 48 that precludes the use of Value Engineering in software acquisition. (12)

3. Value Engineering applies to software when its purpose is achieving the essential function at the lowest life-cycle cost consistent with required performance, reliability, quality, and safety. (11)

4. In FAR 52.248-1(b)(1)&(2), a Value Engineering Change Proposal (VECP) must: Require a change to this, the instant contract to implement; and (2) results in reducing the overall projected cost to the agency without impairing essential functions or characteristics. (5)

5. Value Engineering applies to software when it is viewed as a process. (5)

Surveys which indicated Value Engineering could not be applied to software provided the following responses (frequency of remark is indicated in parenthesis):

1. Software has no recurring cost in production. Software is produced only once, there are no production runs for software, only redevelopment. (4)

2. Software development is considered as R&D, or Architect-Engineering (A&E) which is not allowed in FAR Parts 48.001(b)(2) and 48.104-1(c). (2)

3. There is no method known to quantify Value Engineering savings in software development. (2)

4. The Government does not buy the software, it essentially buys the processes and methodologies that are based on the Software Engineering Institute's Capability Maturity Model (CMM). The five levels in the CMM are used by the DOD to gauge software development risk. (1)

Analysis: This question was designed to determine whether respondents believed language in the FAR was intended exclusively for hardware items. Initial research investigation using telephone queries indicated a majority of contracting personnel did not believe, or had never considered that, Value Engineering could be applied to software. A survey research methodology was determined to be the best course of action to determine what specific elements of FAR Part 48 could be viewed to preclude the use of Value Engineering methodologies in software development.

It is interesting to note the differences in the favorable and unfavorable responses to this question. Among the favorable responses, there was widespread acceptance of the Value Engineering/software relationship contained in the language of the FAR Part 48. The surveys consistently reinforced the idea that Value Engineering, as outlined in the FAR, could be applied to anything whether it was software or hardware.

Among the unfavorable responses, a cultural software development theme from a business perspective was apparent in various forms. For example, many software acquisitions are done in a competitive environment where a Value Engineering program requirements clause is not required. In this situation, the existence of competition alone is sufficient to guarantee the Government the lowest possible cost. Another example is provided in the Capability Maturity Model (CMM) where a contractor's software development proficiency is rated on one of five possible levels, five being the best. The CMM for software provides software engineers with an organized strategy for process improvement. The CMM focus on process improvement sounds very similar to the concepts of Value Engineering. However, the name "Value Engineering" is not specifically identified as such within the scope of process improvement in the software engineering environment.

2. Question Two

The U.S. Army has applied Value Engineering to software reuse. What unique characteristics of software reuse exist that are applicable to the methodologies of VE? Should VE be applied to other areas such as commercial-off-the-shelf (COTS) software products?

__Response__: This question was designed to assess the legitimacy of Value Engineering methodologies in software reuse. This question was complicated by the current legal issues, such as intellectual property rights and liability concerns, which currently plague effective software reuse in DOD. Fifty nine percent of the respondents indicated that the methodologies of VE could be could applied to software reuse. Fifteen percent of the respondents did not believe the methodologies of VE could be applied to software reuse. Twenty six percent of the respondents did not provide a response.

Those that responded favorably provided one or more of the following responses (frequency of remark is indicated in parenthesis):

1. VE can be applied to software reuse if it reduces life-cycle costs and improves reliability. (15)

2. Reuse of software to reduce the number of lines of code to be written is a candidate for VE incentives. (9)

3. Value Engineering methodologies can be applied to anything, therefore, software reuse and COTS are valid candidates. (7)

4. Value Engineering can be used to provide alternatives; software reuse and COTS can be considered suitable alternatives. (2)

5. Value Engineering can be a valid solution for software reuse and COTS provided the measured effort to develop a new application does not exceed what would have been required to develop the original software requirement. (1)

Those that responded unfavorably provided one or more of the following responses (frequency of remark is provided in parenthesis):

1. Value Engineering does not apply to COTS since it is commercial in nature. (7)

2. The practice of using software reuse in DOD has not matured enough to effectively use Value Engineering methodologies. (3)

3. COTS by definition is acceptable as is; changes cannot be made in the name of Value Engineering to a COTS product. (2)

4. Potential software reuse and COTS solutions are required by MIL-STD-498 to be addressed prior to contract awards. (1)

Analysis: A majority of the favorable responses shared a common indepth understanding of the methodologies of Value Engineering. This indication was reinforced by some reference that the respondents had some previous experience or knowledge relating to Value Engineering and expressed unequivocally that the methodologies of Value Engineering did apply to software reuse. It was interesting to note that some of the favorable responses expressed an urgency or necessity to incentivize software reuse with an appropriate contracting tool such as Value Engineering in order to fulfill the potential savings reuse has to offer DOD.

Despite the fact some respondents suggested there have been studies in the software engineering community to determine how Value Engineering applies to software reuse, the submission of VECPs for reuse are virtually non-existent. Based on some of the suggestions presented in the surveys, this reflects an indication that the utilization of software reuse is not yet widely accepted within DOD.

3. Question Three

Some quality characteristics of software include, but are not limited to, understandability, portability, maintainability, testability, and usability. To what extent can the methodologies of VE be applied to these characteristics?

Response: A software product developed with superior software engineering exhibits all the quality characteristics listed above. However, if a software product lacks quality characteristics it may not sufficiently meet the requirements of the user [Ref. 3: P.3-3].

This question was presented to determine the extent to which a VE emphasis on any particular software quality characteristic may contribute additional value to a user in the development of software. Sixty percent of the respondents indicated that VE methodologies could be applied to quality software characteristics; 24% indicated that VE methodologies did not apply to quality software characteristics; and 16% provided no response to the question.

Those that responded favorably provided one or more of the following responses (frequency of remark is indicated in parenthesis):

1. VE methodologies can be applied to the extent that development and life-cycle cost are reduced. (18)

2. VE increases functionality due to improved processes which results in decreased effort and code reduction. (9)

3. VE can be applied to anything, particularly quality characteristics which are most important to the user and can be measured. (9)

4. VE methodologies can enhance portability and maintainability quality characteristics due to longterm considerations affecting life-cycle costs. (8)

5. VE applications provide alternative solutions and tradeoffs by analyzing basic functions and overall design structures. (4)

Those that responded unfavorably provided one or more of the following responses (frequency of remark is provided in parenthesis):

(1) VE savings must be quantifiable in the software environment. Calculating accurate measures of savings are too difficult to determine. (6)

(2) VE has no application since software quality [illegible] specifications and operational requirements documentation. (2)

(3) VE methodologies do not relate to software quality characteristics. (2)

(4) VE is geared to production unit cost; software production cost is insignificant in this case. (1)

Analysis: Since software quality characteristics indicate the extent to which good software engineering takes place during development, it appears logical that some characteristics might be more relevant than others in terms of VE.

A common theme among the favorable responses indicated that all of the quality software characteristics were theoretically applicable to the methodologies of VE. It is interesting to note that maintainability and portability have been specifically identified as having direct VE applications. Figure 8 shows the rising cost of software maintenance relative to software development.

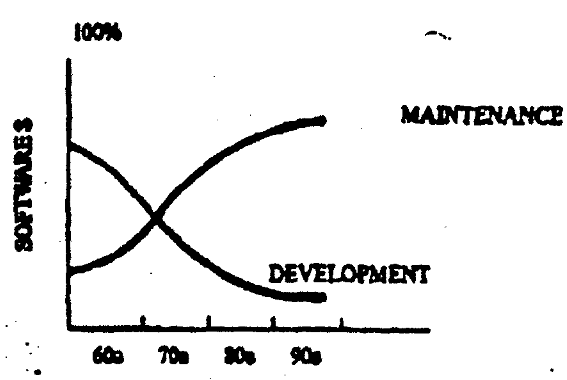

Figure 8 From Ref [21]

Based on the survey responses and the numerous considerations and alternatives discussed in Chapter III, it would suggest VE is a valid methodology for software maintenance. The portability characteristic also lends itself well to VE since it applies to two basic questions that are addressed in any VE study: (1) What else would do the job? and (2) What would the alternative cost?

Measuring the value of quality software characteristics is subjective in nature because there is no single quality measure [Ref. 3: P.3-1]. From a contracting perspective, it is apparent from the FAR that VE is not intended for measures of subjectivity. The FAR emphasizes a tangible "unit cost" for production. Because software is not considered a tangible product, the difficulty in placing a tangible VE savings becomes real and validates the usage of a Cost-Plus-Award-Fee contract where subjectivity is the basis of measurement.

4. Question Four

The software development process includes structured phased/steps. Can the methodologies of VE be applied to any phase in the software development process? Are there any particular phases that do not apply?

<u>Response</u>: This question was designed in order to determine the applicability of Value Engineering beyond the restrictive language of the FAR which emphasizes unit cost. The question was also intended to provide an indication of how familiar software engineers are in the area of Value Engineering. Sixty four percent of the respondents indicated that the methodologies of Value Engineering applied to the software development phases. Seventeen percent of the respondents indicated that the methodologies of Value Engineering did not apply the to the software development phases. Twenty two percent provided no response to the question.

Those that responded favorably provided one or more of the following responses (frequency of remark is indicated in parenthesis):

1. VE applies to any phase of software development that will initiate changes to a contract and will lower total life-cycle cost. (35)

2. VE produces greater savings when implemented in the earliest stages of software development. (15)

3. VE can be applied if new techniques provide new tools or processes that ultimately reduces software development costs. (7)

4. VE is best applied when a baseline or configuration has been established. (3)

5. VE studies can optimize software development processes by tracing and analyzing overall mission requirements. (1)

Those that responded unfavorably provided one or more of the following responses (frequency of remark is provided in parenthesis):

1. VE does not apply to software development. DOD is actually purchasing the most optimal software development process available. (3)

2. Software development is considered R&D. FAR Part 48 specifically excludes R&D efforts. (2)

3. Economical advantages cannot be accurately defined in the phases/steps of software development. (1)

4. Definitions common to VE are intended for hardware applications. The Software Engineering discipline uses unique definitions that do not translate well with "hardware" applications. (1)

Analysis: DOD software engineers have sufficiently demonstrated a strong working knowledge of Value Engineering. The responses strongly suggest that the concepts of Value

Engineering can be applied to software development phases. Numerous responses indicated basic VE applications such as "early VE involvement". Other responses noted that VE changes which result in "reduced life-cycle cost" clearly indicate a positive VE/software development relationship.

However, there is general agreement that any savings incurred as a result of Value Engineering techniques are subjective in nature. Furthermore, these savings are extremely difficult to measure with accuracy.

Since DOD uses the rating levels defined in the Software Engineering Institute's Capability Maturity Model as source selection criteria, any changes in what is already perceived as an optimal software development process is viewed with a high degree of skepticism. Changes to the software development process tend to have a high potential for cost growth. As a result, suggested changes attract considerable management attention in order to control sensitive budget constraints.

5. Question Five

What can or should be done within DOD, if anything, to encourage the use of VE in software acquisition/development (e.g. education, award programs, designate Government savings for use in generating additional savings incentives for contractors, etc.)?

__Response__: This question was designed to determine if DOD VE program initiatives are effective in their current form. Survey respondents were presented with the opportunity to justify the exclusion of VE in software if applicable. Twenty three percent of the respondents did not address the question while 9% indicated no need to encourage additional incentives.

Sixty eight percent provided recommended changes to encourage improvements.

Those that responded with recommendations to improve VE program incentives provided one or more of the following responses (frequency of remark is indicated in parenthesis):

1. Additional emphasis on education is required. (25)

2. Award procedures need improvements to incentivize contractors. Profit is the bottom line. (10)

3. Evolving environmental/cultural changes will induce the required VE program changes. (6)

4. Incentivize VE by encouraging software reuse/COTS utilization. (6)

5. Strong leadership commitment is required to make the VE program work effectively in software acquisition. (5)

6. VE must be emphasized in the contract. (5)

7. Revise FAR Part 48 to incorporate software acquisition. (4)

8. Simplify the VECP submission process. Excessive requirements discourage VECP submissions. (3)

Among those that responded with unfavorable comments provided one or more of the following responses (frequency of remark is provided in parenthesis):

1. A paradigm shift (away from production unit cost reduction) must be addressed before VE can be effective in software acquisition. (2)

2. VE will add unnecessary/excessive bureaucracy to software acquisition. (1)

3. Award Fees are more appropriate than VE incentives in software acquisition. (1)

4. VE is simply not addressed in software development/acquisition. (1)

Analysis: It is interesting to note the favorable recommendations listed above reflect the standard requirements (e.g education, leadership commitment, and streamlining, etc.) of any successful DOD program. The U.S. Congress has documented the same recommendations in order to improve the effectiveness of VE in Federal Agencies. The favorable responses share many of the same ideas for improving VE in software acquisition. These recommendations are typically easy to identify at the working level. However, they are also unusually difficult to implement and manage without consistent leadership and oversight. [Ref. 16]

The unfavorable responses consistently expressed a need to measure VE savings accurately. Another common theme was to eliminate bureaucracy in the form of reducing the number of reviews and to focus on the Software Engineering Institute's CMM to save money.

Approximately one half of the 23% of the respondents that did not address the question simply responded to unique software areas they thought were important. These respondents were from individuals with various engineering backgrounds. No indication was provided whether they had previous contracting experience or otherwise felt unprepared to respond.

6. Question Six

Additional Comments (?):

Response: This question was intended to encourage respondents to provide relevant information about issues or concerns which could be addressed outside the framework of the survey. Thirty three percent of the respondents provided one or more of the following comments (frequency of remark is indicated in parenthesis):

1. The biggest problem in applying VE to software acquisition is quantifying savings. (5)

2. VE is important to software from the maintainability viewpoint.(4)

3. The basis of VE is that anything has the potential to be Value Engineered, including software processes. (3)

4. Configuration management can become a problem due to changes in user requirements. These changes can be cost prohibitive. (3)

5. Software has an integral relationship with computer hardware which is not often displayed in Government and contractor organizations. This hinders good development processes including VE. (2)

6. Everyone in DOD thinks software development is "black magic," and it is not. Software is no different than any other discipline, except that it is new. Neither is it easy. (1)

Analysis: The difficulty in accurately quantifying VE savings in software development is a common concern among software engineers. However, no responses indicated that quantifying VE savings could not be done. Maintaining software is another area of shared concern among software engineers because the amount of fielded software is growing. As discussed in Chapter III, this is a valid concern. It costs more to maintain software once the initial design has been completed. [Ref. 9]

It is interesting to note that this was the only question that prompted the connection between hardware and software. Since software is dependent upon hardware, it would seem logical that a VE software study would require corresponding hardware components to undergo some scrutiny to determine if alternatives are available.

C. SUMMARY

This chapter reported and analyzed the perceptions of Program Managers, software engineers, and contracting specialists regarding the application of Value Engineering methodologies to software acquisition/development. A significant majority provided favorable responses to questions one through five. It appears that numerous Value Engineering opportunities and significant savings could be achieved if the current DOD Value Engineering program was modified to accommodate software acquisition.

Various suggestions were made that could facilitate the required changes. However, based on the responses submitted a significant paradigm shift will be required to implement Value Engineering methodologies. For example, 4% of the respondents specifically stated that after years of software engineering experience, they had never observed VE applications in software development. Three percent indicated that they used VE methodologies in software development. However, there were only <u>four examples</u> that could be recalled by software engineers where software VECPs were documented in the last five years. In one case, an Air Force contractor had to be made aware of the VE opportunity and even encouraged to submit the VECP. This suggests that Value Engineering methodologies need additional management emphasis in order to be effective.

V. CONCLUSIONS AND RECOMMENDATIONS

A. OVERVIEW

Since the latter part of the 1980's, the Department of Defense (DOD) budget has been shrinking. Value Engineering has been an effective cost saving technique for both Government and Industry alike. In a prepared response to the latest update to OMB Circular A-131, Dr. John Deutch stated,

> The DOD Value Engineering (VE) program, through our internal and industry efforts, reports savings and cost avoidance of over $1 billion annually, more than any other DOD cost reduction program. [Ref. 7]

This statement eloquently demonstrates that Value Engineering is a worthy program which warrants continued support well into the future. As DOD approaches the 21st century, senior management will undoubtedly look to the most effective cost saving programs available. Value Engineering is one possible solution to minimize the negative effects of downsizing the military establishment. However, the success of a program such as Value Engineering will ultimately depend upon DOD's ability to manage the cultural and political challenges of changing environments and smaller budgets.

The focus of this research was to determine how the Department of the Navy manages its Value Engineering program with respect to computer software acquisition, and to what extent the methodologies of VE could be utilized to reduce ever increasing computer software costs. This chapter will present the conclusions and recommendations of this research effort.

B. CONCLUSIONS

1. The Federal Acquisition Regulation (FAR) Part 48 does apply to software acquisition. However, it was written with an emphasis on hardware and unit cost reduction.

FAR Part 48 makes numerous references to "unit cost" which are associated to the various definitions relating to acquisition savings. A term such as "unit cost" does not lend itself well to software acquisition. "Unit cost" tends to relate to tangible items such as hardware. Software is not considered tangible.

Twenty seven percent of the survey responses specifically stated that VE does apply to software only because of the general definition of VE characterized in FAR 48.101. Beyond the general definition of VE, there is no specific reference that discusses the calculation of acquisition savings associated with software VECPs. As currently written, FAR Part 48 provides no guidance whatsoever in applying an accurate savings formula to software acquisitions.

2. The methodologies of VE do apply to computer software development and acquisition.

This conclusion is based on the nature of Value Engineering. Recall from Chapter II that VE is a philosophy and not an exact science. Anything can be "Value Engineered", regardless of whether it involves hardware or software. Value Engineering challenges everything and excludes nothing to identify inefficient cost. Until an item or process has been declared "absolutely perfect", then VE can always be utilized to achieve improvement. Mankind will always continue to strive to "build a better mousetrap."

In software development, there are several development processes and tools available that assist the software engineer. Some processes and tools are better than others, but all are subject to improvement. It is important to keep in mind that as a discipline, software engineering is still relatively new. Since the 1960's there have been incredible advances in software developments which have tremendously increased the capability of every major weapon system. Over the next 30 years, the methodologies of VE can be used to accelerate the development of software to unprecedented levels of performance that were previously thought to be impossible.

3. DOD software acquisition policies do not effectively support the utilization of VE.

We have seen from Chapters III and IV that software reuse and COTS can be considered valid candidates for VE. However, the military standard for development and documentation, MIL-STD-498 precludes the use of VE for these applications. Recall that contractors are required to identify potential reuse/COTs solutions in their Request for Proposals. This virtually eliminates any possibility to employ VE solutions in software development after the contract has been awarded.

Another area of software acquisition that precludes the use of VE in awarding a contact is based on the Software Engineering Institute's (SEI) Capability Maturity Model (CMM). DOD contractors can have their software development processes rated by SEI. The ratings in the CMM range from one to five levels; five being the best. When DOD contracts for software, it does not procure a tangible item, it procures the contractor's software development process to minimize risk associated with the complexity of the contract's requirement.

To be competitive, a contractor must continuously improve their software development processes in order to progress to level 5 in the CMM. One could argue that the process improvements contractors focus on to be competitive is actually Value Engineering. Unfortunately, the term Value Engineering is not associated with the CMM.

To that end, the term Value Engineering is not found in any DOD software publication or guideline. There are of course processes that can be "Value Engineered" in DOD software development, but these processes do not define the term as Value Engineering.

4. Contracting personnel and Program Managers (PM) require additional training in software development.

Virtually everyone this researcher talked to regarding software VECPs expressed common difficulties in submitting VECPs. There are two significant problems involved, and both are related to each other. The first problem deals with educating contracting personnel. The second problem deals with the degree of difficulty required to get a software VECP approved.

a. *Education*

To successfully implement any VECP, contracting personnel must possess a basic understanding of the change being considered and how that change affects the contract. In the area of software development, contracting officers and PMs do not have the education to properly manage major weapon system contracts that are software intensive.

In February 1995, the DOD Software Acquisition Best Practices Initiative Workshop in Warrenton, Virginia identified this deficiency as a significant management problem in software development. Various speakers at the Workshop reported two significant findings: (1) education problems associated with contracting officers result in poor contract administration because problems in software development cannot be anticipated, identified, and/or corrected in a timely manner, and (2) PMs need additional education in order to provide the Defense Acquisition Board valid information during Milestone reviews.

With the problems discussed above, this causes contracting officers and PMs to view software related VECPs with suspicion. A VECP will not get approved unless it is thoroughly understood by contracting officers and PMs. One survey response specifically stated that a software related VECP was disapproved because it was not believed that VE applied to software.

b. VECP Avoidance

Because contracting officers and PMs do not have adequate training in software, software engineers will avoid VECPs as a contract incentive or requirements solution. Survey respondents who indicated previous experience in VE stated there is too much effort required to submit and follow up on software VECPs. As a result, software engineers will seek alternative methods to fulfill their meet objectives.

5. Implementing VE methodologies in software development and acquisition will require dedicated management commitment to achieve acceptable levels of success.

Value Engineering and software engineering are two different disciplines, particularly in DOD. This research concludes that there is no administrative mechanism currently available to connect them. There is no written instruction or guidance in DOD which specifically directs the use of VE in software acquisition.

Without such guidance or instruction, it is incumbent upon management to provide the leadership to make the necessary changes. Simply put: personnel in non-management positions cannot be expected to effectively incorporate drastic change without proper guidance from leadership.

C. RECOMMENDATIONS

1. Any attempt to incorporate VE in software acquisition will first require a comprehensive analysis. Senior DOD management will need to determine the feasibility of such a change.

As discussed earlier, there are no administrative mechanisms available in DOD to connect Value Engineering and Software Engineering. A comprehensive analysis would be required to determine what impact or influence the introduction of VE would have on software acquisitions. At a minimum, current guidance such as the new MIL-STD-498 would require considerable changes. In light of the dynamics of software acquisition, dramatic changes in basic procedure would in all likelihood be extremely unpopular. Most people in Government naturally resist additional program requirements thrust upon them, regardless of the circumstances.

In any event, a significant "paradigm shift" would have to occur in both industry and Government to incorporate VE in software acquisitions. This would obviously take time to gain acceptance. It would also take time to learn how to make the environment of software amenable to VE. However, if a study concluded VE should be incorporated in software acquisition, then the following sub-recommendation would apply:

a. *Revise FAR Part 48*

This section of the FAR was written at a time when DOD was experiencing a military buildup. Today, DOD budgets have constrained the number of weapon systems that are being purchased. DOD simply cannot buy the number of tanks, ships, and planes that it did in the 1980's. A FAR revision is in order to accommodate smaller acquisition quantities. This revision will also have to address simple procedures for calculating savings associated with software VECPs.

2. Determine and provide adequate software acquisition and development training to contracting officers and Program Managers.

As discussed earlier, contracting officers and PMs do not have adequate training in software. It only makes good sense to require acquisition personnel to have more than a basic understanding of what they are buying. Every major weapon system in DOD is software intensive, therefore, no contracting officer or PM can avoid software related procurements. Recall from Chapter 1 that in fiscal year 1994, software costs for DOD were $42 billion. With so much time and money being spent on software, contracting officers and PMs should be adequately trained to manage the administrative difficulties associated with software acquisition.

D. RESEARCH QUESTIONS

The following subsidiary research questions were germane to the research effort:

<u>What are the principal features of the U. Navy's VE program</u>

The principal features of the Navy's Value Engineering program are based on various acquisition guidelines which incorporate the policies discussed in the FAR. To implement the regulations Congress outlines in the FAR, the Executive Branch of Government issued OMB Circular A-131 which directs Federal Agencies to use VE as management tool where appropriate. In DOD, the principal features of VE are outlined in DOD Instruction 5000.2 These features essentially define VE in DOD and provide requirements for reporting annual VE activity statistics for each DOD component. The VE report is used to gauge the status of the VE program and to identify areas of improvement.

Within the U.S. Navy, each Systems Command promulgates its own instruction to establish a Value Engineering program. These command instructions provide specific guidance in training and reporting procedures. Specific staff positions are identified and explicit VE responsibilities are discussed. Field activities assigned to Systems Commands are included as action addressees.

<u>What is the role of the Value Engineering Change Proposal (VECP) and how is it applied to VE?</u>

The VECP is the contractual mechanism that implements VE in a contract. The VECP is used by contractors to document suggestions which encourage a change to a contract. The contractor is essentially attempting to justify a more economical method to fulfill contractual requirements. Therefore, a change resulting from the implementation of a VECP reduces the cost of a contract. The corresponding reduction in cost is then shared between the Government and the contractor based on the share ratios listed in the FAR.

<u>What characteristics, if any, of computer software acquisition are most pertinent to the application of VE concepts?</u>

Value Engineering methodologies can be applied to several aspects of computer software acquisition. By analyzing software engineering concepts such as The Plurality of Goals, and Marginal Analysis, VE studies can provide valuable insight into available alternatives. Each alternative can then be judged according to its perceived value by the user. Ultimately, the user will make a decision that maximizes the value of the acquisition.

Other unique aspects of software acquisition that can apply to VE include topics such as Goldplating and Legacy Systems. These aspects are much easier to understand in terms of VE applications because they have relatively simple concepts that translate well with "hardware".

One objective of Value Engineering is to reduce total life-cycle costs. This research demonstrated that VE in software maintenance applications can have a significant potential to reduce total life-cycle costs. Similarly, software reuse and COTS applications were also shown to be valid VE candidates.

<u>How do U.S. Navy contractors and in-house personnel view the concept of VE with regard to computer software acquisition?</u>

Survey results in this research indicated that a majority of these people believe that VE does apply to software. However, there is no administrative mechanism available to connect VE and software acquisition. While contractors all agree they strive to continuously improve their development processes, it is clearly recognized that it is not being done in the name of VE.

Value Engineering must be emphasized repeatedly in the area of software development in order to be effective. This includes the insertion of VE references in all DOD software [illegible] software relationship. No DOD program succeeds without solid backing from management. It is clearly not sufficient to make **a one line reference** that states VE applies to software in OMB Circular A-131 and DOD Instruction 5000.2. There is much more that can be done in order to inundate VE in software acquisition.

What approach, if any, should the U.S. Navy use to facilitate the application of VE/VECPs to computer software acquisition?

Education is the key that will facilitate the application of VE/VECPs in computer software acquisition. Contracting officers must be trained to identify unique opportunities where VE can apply to software. As with any VE application, contracting officers must aggressively seek out VECP opportunities presented by contractors. To be successful, this will only happen if contracting officers have been properly trained. Value Engineering is done because it makes good sense. Accordingly, contracting officers must know what makes good sense in software acquisition.

The primary research question for this study was: How, and to what extent can the Department of the Navy's Value Engineering Program be utilized in the acquisition of computer software?

This research has demonstrated that Value Engineering can be utilized in the computer software acquisition. This can be accomplished by applying the methodologies of VE to the following concepts of software engineering:

(1) The Plurality of Goals.

(2) Marginal Analysis.

(3) Goldplating.

(4) Software Maintenance.

(5) Legacy Systems-Business Value Analysis.

(6) Software Reuse/COTS.

Although there are opportunities available to utilize VE methodologies in software acquisition, there is very little evidence to suggest that the U.S. Navy takes advantage of those opportunities. This research has identified two distinct reasons which directly contribute to the lack of VE in software acquisition.

Contracting Officers and Program Managers lack adequate training and education in software to effectively implement VE in software acquisition. To add to this problem, the DOD Value Engineering program as a whole has had a history of management and visibility problems [Ref 34: P.17]. Without adequate training and education, the U.S. Navy's VE program cannot be effective in software acquisition.

There is no administrative mechanism that connects VE and software acquisition. OMB Circular A-131 and DOD Instruction 5000.2 merely state in one sentence that VE applies to software. FAR Part 48 makes no reference whatsoever to software. Furthermore, there is no specific U.S. Navy guidance that addresses a recommended approach to exploit VE opportunities in software acquisition.

With the lack of VE emphasis in software development and acquisition, the focus on improving software development and acquisition processes are defined in MIL-STD-498 and the SEI Capability Maturity Model. Both of these software guidelines are currently the preferred tools that contractors rely upon to concentrate on continuous improvement. By focusing on continuous improvement in the development of software, contractors can readily gauge their relative competitive position in the source selection process. This continuous improvement paradigm in software development currently obviates the need for VE.

E. AREAS FOR FURTHER RESEARCH

The following area is recommended for further research:

Conduct an analysis of the top ten DOD contractors to determine what changes, if any, are currently being implemented in their VE programs. DOD has witnessed several defense contractors merge within the last five years. These mergers are occurring in the name of competition and survival. A useful study would concentrate on specific elements of a "new contractor's" VE program to determine which elements are successful and why.

This analysis would necessarily investigate a contractor's VE approach to subcontracting plans. The result a this study could shed light on how to effectively manage a VE program for subcontractors. DOD could use this information to efficiently manage reduced budgets and to assist other contractors in the hopes of keeping them competitive.

In any event, this analysis could also provide insight into an effective management approach to VE. DOD will be procuring fewer weapons in the future and that will tend to reduce the VE opportunities that were available in the past. This is the right time to development new VE approaches in DOD. We simply cannot afford to do business in the future by looking at the past.

APPENDIX

SURVEY QUESTIONS

1. The Federal Acquisition Regulation (FAR) part 48 discusses Value Engineering (VE) requirements. In your opinion, does VE apply to computer software? If your answer is no, please state specifically what subpart of FAR part 48 precludes the use of VE in computer software acquisition and why. If your answer is yes, please state specifically what subpart of FAR part 48 does apply to software acquisition and why it does.

2. The U.S. Army has applied Value Engineering to software reuse. What unique characteristics of software reuse exist that are applicable to the methodologies of VE? Should VE be applied to other areas such as commercial off the shelf (COTS) software products?

3. Some quality characteristics of software include, but are not limited to, understandability, portability, maintainability, testability, and usability. To what extent can the methodologies of VE be applied to these reuse characteristics?

4. The software development process includes structured phased/steps. Can the methodologies of VE be applied to any phase in the software development process? Are there any particular phases that do not apply?

5. What can be done within DOD, if anything, to encourage the use of VE in software acquisition/development (ie. education, award programs, designate Govt. savings for use in

generating additional savings incentives for contractors, etc.)?

6. Additional Comments (?):

LIST OF REFERENCES

1. Bennett, Keith, "Legacy Systems: Coping With Succuss," IEEE Software, January 1995, pp.19-21.

2. Birrell, N.D. and M.A. Ould. A Practical Handbook for Software Development, New York: Cambridge University Press, 1985.

3. Boehm, Barry W., John R. Brown, Hans Kaspar, Myron Lipow, Gorden J. Macleod, and Michael J. Merrit. Characteristics of Software Quality, North-Holland Publishing Company: New York, 1978.

4. Boehm, Barry W., Software Engineering Economics, Englewood Cliffs, NJ: Prentice-Hall Inc., 1981.

5. Cash, James I. Jr., Warren F. McFarlan, James L. McKinney, and Lynda M. Applegate. Corporate Information Management: Text and Cases, Third Edition, Boston, MA: Irwin, 1992.

6. Department of Defense MIL-STD-498, "Military Standard; Software Development And Documentation," 5 December 1994.

7. Deutch, John M., "Office of Management and Budget (OMB), Value Engineering," Under Secretary of Defense Memorandum, 10 December 1993.

8. Dobler, Donald W., David N. Burt, and Lamar Lee Jr., Purchasing and Materials Management, New York: McGraw Hill, Inc., 1990.

9. Editors of The Defense Systems Management College, Mission Critical Computer Resources Management Guide, 1990.

10. Eliot, Lance B., "Legacy Systems, Legacy Options," Computer World, 11 July 1994, p.88.

11. Federal Acquisition Regulations, Part 48, Chicago: Commerce Clearing House, Inc., February 1992.

12. Given, Gary, G. Government Management of Contractor Submission of Value Engineering Change Proposals, Thesis, Naval Postgraduate School, December 1985.

13. Glaseman, Steven, Comparative Studies in Software Acquisition, Lexington Books, D.C. Health and Company: Lexington, MA. 1982.

14. Green, Robert, "Army Has a Carrot for Software Contractors," Government Computer News, 25 November 1991, p.38, col. 1.

15. Headquarters Air Force Logistics Command, "Air Fore Value Engineering Program," AFLC Supplement 1, AFR 320-1, 30 May 1986.

16. Hearing Before the Subcommitte on Oversight of Government Management of the Committee on Government Affairs United States Senate, "Oversight of Value Engineering Program in Federal Agencies," Senate Hearing No. 100-203, 29 April 1987.

17. Horngren, Charles T., George Foster, and Srikant M. Datar, "Target Costing for Target Pricing," Cost Accounting 8th Edition, Englewood Cliffs, NJ: Prentice Hall, 1994.

18. Lamm, David V. and Pursch, William C., "A Dictionary of Contracting Terms Part III," Contract Management, April, 1993, p.15.

19. Miles, Lawrence D., Techniques of Value Analysis and Engineering, New York: McGraw-Hill Book Company, 1961.

20. Naval Information Systems Management Center, Software Acquisition Best Practices Initiative Workshop, Warrenton, Virginia, 13-17 February 1995.

21. Paley, Alfred I., and Joseph H. Jones. "Applying VA/VE To Computer Software Engineering." Save Proceedings, 1992. pp. 269-270

22. Perry, William J., "Specifications and Standards--A New Way of Doing Business," Secretary of Defense Memorandum, June 1994.

23. Report of the Software Acquisition Management Education Review Team, A Report for the Deputy Under Secretary of Defense (Acquisition Reform), 7 March 1994.

24. Shank, John K., and Vijay Govindarajan, "Concepts in Value Chain Analysis," Strategic Cost Analysis, Boston: Irwin, 1989.

25 Sneed, Harry M., "Planning the Reengineering of Legacy Systems," IEEE Software, January 1995, p.28.

26. Software Reuse Initiative Program Management Office, "Reuse Executive Primer-Working Draft," Third Edition, 8 February 1995.

27. Webster's II New Riverside University Dictionary, Boston, MA: The Riverside Publishing Company, 1984.

28. U.S. Department of Defense Directive, "Defense Acquisition Management Policies and Procedures," DODD 5000.2, February, 1991.

29. U.S. Department of Defense, "Value Engineering," DOD INSTRUCTION 4245.8H, March 1986.

30. U.S. Department of the Navy, "Value Engineering Program," COMNAVAIR INSTRUCTION 4858.3B, 13 January 1994.

31. U.S. General Accounting Office, "Embedded Computer Systems, Defense Does Not Know How Much It Spends on Software," GAO/IMTEC-92-62BR, July 1992.

32. U.S. General Accounting Office, "Mission Critical Systems; Defense Attempting to Address Major Software Challenges," GAO/IMTEC-93-13, December 1992.

33. U.S. General Accounting Office, "Software Reuse; Major Issues Need to Be Resolved Before Benefits Can Be Achieved," GAO/IMTEC-93-16, January 1993.

34. U.S. General Accounting Office, "Value Engineering; Usefulness Well Established When Applied Appropriately," GAO/T-GGD-92-55, 23 June 1992.

35. U.S. Office of Management and Budget, "Value Engineering," Circular NO. A-131, 21 May 1993.

36. 104th Congress, House of Representatives Bill 719, Systematic Application of Value Engineering Act of 1995, 27 January 1995.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 52 Naval Postgraduate School Monterey, California 93943-5101	2
3.	Defense Logistics Studies Information Exchange U.S. Army Logistics Management College Fort Lee, Virginia 23801-6043	2
4.	CDR Jeffery Warmington, SC, USN, Code SM/Wr Naval Postgraduate School Monterey, California 93943-5002	1
5.	Dr. David Lamm, Code SM/Lt Naval Postgraduate School Monterey, California 93943-5002	5
6.	LTC Robert D. Buckstad PM Avionics, SFAE-AV-AEC 4300 Goodfellow Blvd St. Louis, Missouri 63120-1798	1
7.	Mr. Merrill Yee Office of the Assistant Secretary of Defense (Economic Security) OASD(ES)PR/IEQ/IQPD Industrial Quality & Productivity Div. 5109 Leesburg Pike, Suite 304 Falls Church, Virginia 22041-3466	3
8.	LT Charles T. Race 36 Evergreen St. Cortland, New York 13045	2
9.	LT David C. Meyers 310 Arlon Ct. Seaside, California 93955	1