114
73

OBJECTIVES, PRINCIPLES, AND ATTRIBUTES: AN APPROACH TO METHODOLOGY DEVELOPMENT AND EVALUATION FOR SYSTEMS ENGINEERING

by

David K. Kreider

Project report submitted to the Faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

MASTER OF ENGINEERING

in

Systems Engineering

APPROVED:

Richard E. Nance, Chairman

James D. Arthur

Benjamin S. Blanchard

Stephen W. Parker

January, 1990

Blacksburg, Virginia

OBJECTIVES, PRINCIPLES, AND ATTRIBUTES:
AN APPROACH TO METHODOLOGY DEVELOPMENT AND EVALUATION FOR SYSTEMS ENGINEERING

by

David K. Kreider

Committee Chairman: Richard E. Nance
Computer Science Department

(ABSTRACT)

The primary purpose of this paper is to propose a future direction of investigation: a more formalized approach to developing systems engineering methodologies and tools to support the system engineer and other disciplines in effectively applying the systems engineering process. This approach is based on the concept of using design principles to achieve specified objectives. The achievement of these objectives is indicated by the presence of defined attributes within the developed system. A secondary purpose of this paper is to illustrate how the concept of objectives, principles, and attributes provides a means for promoting an increased understanding of the similarities and interrelationships between systems engineering and other disciplines, particularly software engineering. A direct consequence of this increased understanding is an improvement in the application of the systems engineering development process.

ACKNOWLEDGEMENTS

My sincere appreciation and thanks is given to my project chairman and advisor, Dr. Richard Nance, for his guidance and insight into approaches to improve the implementation of both the systems engineering and software engineering processes. His direction, personal attention to this project, and understanding of the processes facilitated my maintaining the focus of this paper. I am grateful to Dr. James D. Arthur, Ben Blanchard, and Stephen Parker for serving on my project advisory committee and for offering their valuable opinions and guidance concerning the project.

Special thanks is given to Brian Zagoric and Kevin Kolb for reviewing my paper for grammatical correctness and for providing suggestions for improving the papers understandability.

Most of all, I give my deepest thanks to my family and friends for their support and unending patience and understanding throughout this effort.

TABLE OF CONTENTS

TABLE OF CONTENTS

Listing of Illustrations:

Listing of Tables:

1.0 INTRODUCTION

Systems engineering is a relatively new field with respect to the traditional fields of engineering. Its origins can be traced to military operational planning efforts and problem solving methodology development in the 1940's and the ballistic missile programs of the 1950's. [10, 28, 45] These dates also roughly coincide with the advent of computing technology--a fact that is not incidental. It is the advances in computer technology and digital logic circuitry that have made possible the development of more highly complex systems. [18] As computing technology advanced, the need to incorporate this and other new technology efficiently and effectively became paramount. By the 1960's systems engineering was beginning to be formally recognized and organized as it evolved to address the issues of increasing system complexity and the need for a logical, formalized system acquisition process. [45]

As technology continues to advance and promote the development of larger, more complex systems, systems engineering must continue to evolve to support development efforts. Providing only a mental framework to guide the engineer will eventually be inadequate. Tools, methodologies, and training programs need to be developed and integrated into a structured systems engineering environment supporting the entire system life cycle (Figure 1). One such infrastructure to support future evolution is

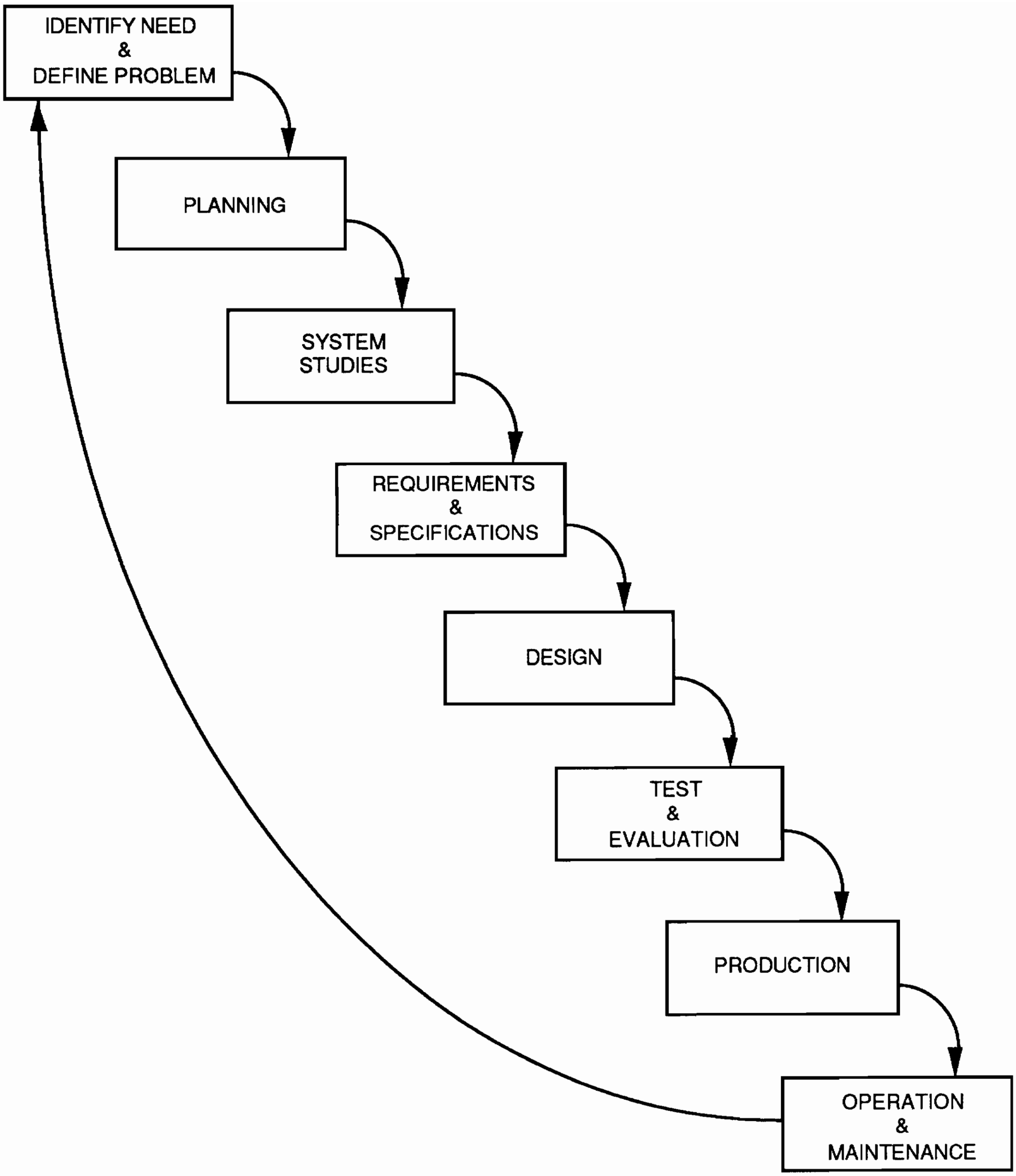

Figure 1. The Systems Life Cycle

the concept of systems engineering objectives, principles, and attributes (OPAs).

This paper focuses on defining the OPA concept, the relationship between systems and software engineering, and illustrating how the OPA approach can improve the application of the systems engineering process and its relationship with other disciplines.

2.0 CONCEPTS AND DEFINITIONS

This section is intended to familiarize the reader with and establish the context of several important terms and concepts as presented within this paper. These terms and concepts are system, systems engineering, software engineering, system engineer, method, methodology, and process. This background material will aid the reader in understanding other concepts presented within the paper.

2.1 SYSTEMS

A system may be defined as a composition of personnel, equipment, resources, and operations (interconnected yet separate and independent) organized to achieve a specified objective. [27, 28] Various types of systems exist with different levels of complexity and magnitude. Systems can be biological, sociological, mechanical, dynamic, static, etc., in nature. [4]

In order to limit the scope of the types of systems addressed within this paper, only the category of systems having the following common properties is considered [25, 27]:

1) A system is a complex, man-made grouping of resources (equipment or hardware, software, and personnel). In being complex, a change in one or more of the resources affects the other resources.

This effect can be adverse, neutral, or beneficial.

2) A system may be decomposed into subsystems.
3) Systems have precisely defined goals.
4) System components have integrity; that is, they contribute to the achievement of a common objective or goal.
5) A system is often a part of a larger hierarchy of systems.
6) Systems can be competitive, cooperative, or independent.

Thus based upon one or more of the above properties, biological or sociological systems are excluded from consideration in this paper.

2.2 SYSTEMS ENGINEERING

Traditionally, the engineering approach to problem solving has been to decompose the problem into smaller, solvable problems. These smaller, individual problems are then analyzed, solutions developed, and solutions selected and synthesized in such a manner as to solve the larger problem. This is often referred to as the conventional approach. [32]

The systems engineering process is based upon the combination of the conventional approach with the concept of expansionism. Expansionism recognizes that individual systems not only are comprised of subsystems but also may be a part of a larger hierarchy of systems. [32] This is reflected in the systems engineering process's synergistic approach emphasizing the "whole" system design rather than the design of the individual subsystems or components which comprise the overall system.

No single universally accepted definition for systems engineering exists. [4, 18, 27] Furthermore, similarities or differences dominate, depending on one's viewpoint. [18] While it is not the purpose of this paper to define the systems engineering process, some definitions are necessary to establish the concept.

MILITARY-STANDARD-499A [31] defines systems engineering as "a logical sequence of activities and decisions transforming an operational need into a description of system performance parameters and a preferred system configuration."

Eisner [18] defines systems engineering as "an iterative process of top-down synthesis, development, and operation of a real-world system that satisfies, in a near-optimal manner, the full range of requirements for the system."

The Defense Systems Management College defines systems engineering as "the application of scientific and engineering efforts to (a) transform an operational need into a description of system performance parameters and a system configuration through the use of an iterative process of definition, synthesis, analysis, design, test, and evaluation; (b) integrate related technical parameters and ensure compatibility of all physical, functional, and program interfaces in a manner that optimizes the total system definition and design; and (c) integrate reliability, maintainability, safety, survivability, human, and other such factors into the total engineering effort to meet cost, schedule, and technical performance objectives." [45]

Jenkins and Youle define systems engineering as the activity of planning, designing, constructing, testing, and operating complex systems. It provides a framework to collaborate the many separate and possibly divergent disciplines which otherwise might fail to make an effective contribution to the overall success of the problem, and to design the best system for any particular objective. Therefore systems engineering is a team activity bringing together a diversity of specialists as well as system engineers. [25]

While not explicitly stated, several similar and common concepts expressed in these definitions deserve comment:

- systems engineering is an iterative process,
- systems engineering occurs over the entire life cycle of a system, [4, 18, 43]
- systems engineering is interdisciplinary in nature, [43]
- system engineers must integrate not only system components but also the technical specialties

involved in the development process, production, maintenance, logistic support, etc.,

- the systems engineering process transforms a need into requirements and specifications, and
- systems engineering emphasizes system optimization over individual subsystem or component optimization.

2.3 SOFTWARE ENGINEERING

While systems engineering is aimed at creating systems that economically meet requirements and are reliable and maintainable, software engineering is aimed at creating software code with similar desirable characteristics.

As is the case in systems engineering, there is no universally accepted definition of software engineering. The following definitions are provided to establish the concept. Nance and Warner [49] reference the Department of Defense Directive 5000.29 in defining software engineering as the science of design, development, implementation, test, evaluation, and maintenance of computer software over its life cycle.

Booch [6] defines software engineering as the discipline providing a consistent, life-cycle approach to the creation of software systems. It aids design of software systems addressing complex issues by providing tools to help manage the complexity.

Fairley [20] defines software engineering as "the technological and managerial discipline concerned with systematic production and maintenance of software products that are developed and modified on time and within cost estimates."

2.4 SYSTEM ENGINEER

A system engineer is a generalist, trained to think in terms of an overall approach to solving problems. Within the systems engineering process, it is the system engineer's responsibility to ensure a correct, complete definition of the need and its transformation into a set of requirements and specifications. Once this is achieved, the system engineers organize the requirements and specifications into units which can be built and then ensures complete "integration" of the components into the preferred system which effectively satisfies the specified need.

In order to accomplish this integration, the system engineer must be able to liaise and communicate effectively with various specialists involved within an interdisciplinary approach to system development (Figure 2). The system engineer has the responsibility of coordinating these different disciplines within a total engineering effort to ensure the objectives are communicated to and understood by all involved in developing the system, and to ensure the

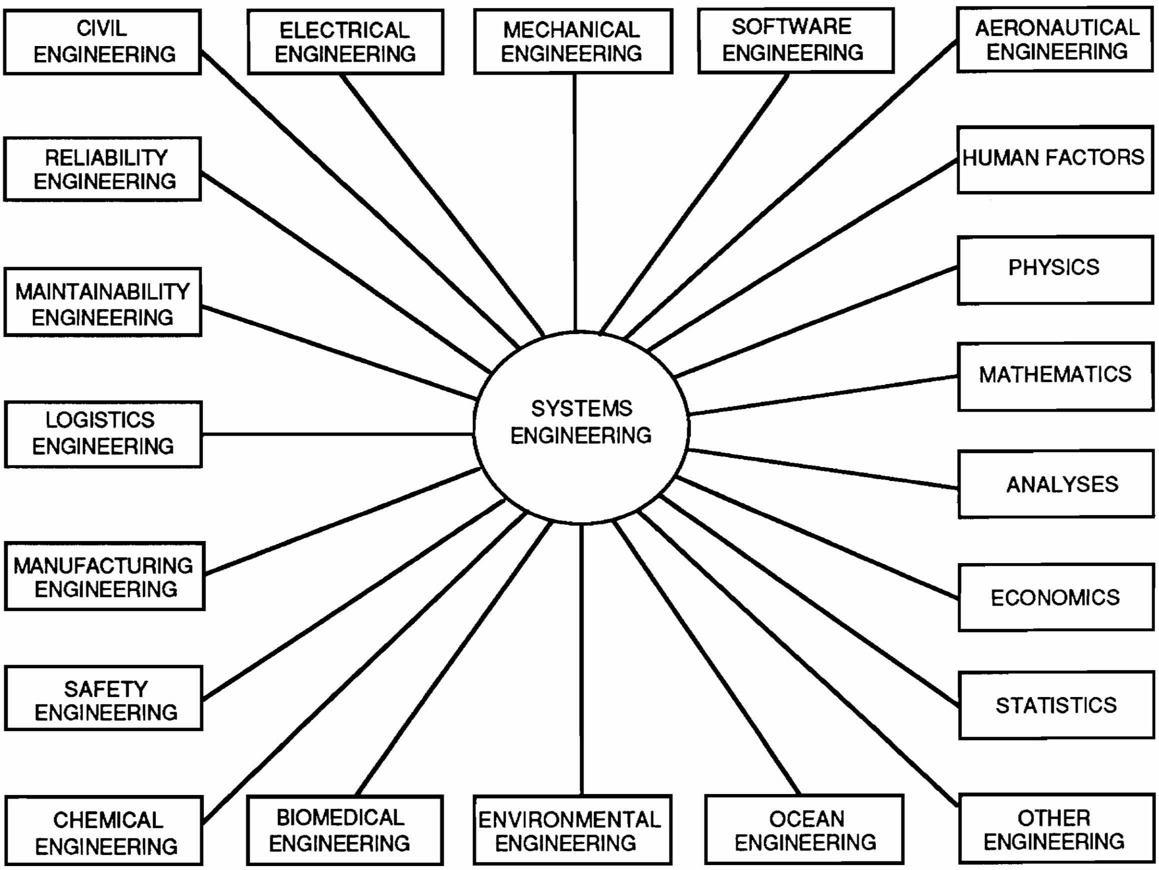

Figure 2. Systems Engineering Integration of Disciplines [43]

agreed upon objectives are realized during the development, operation, and support of the system in order that the system meets the specified operational needs. [4, 18, 25, 41]

2.5 METHODS AND METHODOLOGY

The terms "method" and "methodology" are often used synonymously in engineering literature. Yet, they connote distinctly different meanings. [1] The American Heritage dictionary defines a method as a regular and systematic means or manner of procedures and techniques characteristic of a particular discipline or field of knowledge. In general, a method specifies three aspects of decision making [1, 16]:

(1) the set of decisions to be made,

(2) how they are to be made, and

(3) the sequence in which they are to be made.

In contrast, a methodology is a system of complementary methods and a set of rules and constraints for applying the methods in an organized manner to accomplish a specific task. [1] In general, a methodology

- defines the methods used to achieve the global objectives;
- organizes, dictates, and establishes the relationships among tasks composing the methods

utilized to achieve global objectives; and

- determines the order in which certain classes of decisions are made, and the means of making those decisions leading to the overall desired objectives. [1]

System objectives or goals vary widely; consequently, different methods and methodologies may be employed by organizations to achieve specific objectives of interest to that organization. Conversely, a methodology could be developed to place emphasis on attaining several specific objectives common to several types of systems or of interest to multiple disciplines. Further, it is common for methods or methodologies to be applicable to specific phases or activities within the systems life cycle in contrast to addressing and supporting the entire life cycle. In general, to achieve a desired objective or objectives, the methodologies used should be based on accepted engineering principles. [1, 16, 36]

2.6 PROCESS

The American Heritage Dictionary defines a process as a series of actions, changes, or functions that bring about an end or a result. A process does not specifically tell the user how or when to make decisions. A process provides the type or nature of decisions that are required to be made. Therefore the key distinction between a process and a methodology, as proposed in this paper, is that a process represents a framework or guideline for the series of actions and decisions to be made. A methodology provides specific steps to decision making. That is to say a methodology states the decisions to be made, how to make the decisions, when to make them, and in what order.

Various literature sources claim to present a systems engineering methodology representing the system development process. A tenet of this paper is that the methodologies reviewed to date may be more appropriately termed "processes" which represent the system life cycle. It must be recognized that not all authors of systems engineering literature subscribe to the definitions of methodology and process presented within this paper.

The claim that "process" is a more appropriate term with respect to systems engineering is not meant to be derogatory. The term "process" connotes the flexibility for adaptation to each unique system development effort. Furthermore, these "processes" presented in literature indicate areas for future development of methodologies within the field of systems engineering. To be more specific, methodologies should be developed to support the efforts within a particular phase or, more desirably, the entire life cycle of the systems engineering process. Thus, while systems engineering can be viewed as a process, there is a need for methodologies to aid the system engineer in applying the process.

3.0 THE NATURE OF SYSTEMS ENGINEERING

This section is intended to familiarize the reader with the nature of systems engineering, the various factors influencing the systems engineering process, and the approach taken within this paper to address these factors.

3.1 TECHNICAL COMPLEXITY AND SYSTEMS ENGINEERING

The problems system engineers are expected to address and solve can be exceedingly complex and difficult. The systems engineering process is a logical, structured approach to system design and development employed by the system engineer to contend with the technological and sociological complexity of modern systems. [30]

In a further effort to aid the system engineer in reducing the issue of complexity within modern system development, tools and methods have been and continue to be developed. However, these design methods are often inadequate, particularly with respect to designing systems with high man-machine interfacing. [48] Furthermore, the tools and methods are often not designed to be coordinated and integrated (themselves or their results), nor do they support the entire system development life cycle. [26] As a result they may fall short of their maximum potential benefit to the system engineer.

Another approach to confronting the complexity of system development is the utilization of specialists. [15] The specialist may address specific types of problems, perform specialized design of subsystems, or address specific phases of the life cycle within their area of expertise. Their work is then, in theory, synthesized into the whole system within the systems engineering process. However, use of specialists often introduces communication problems which are discussed in a subsequent section on systems engineering vulnerability points.

Considering the various approaches utilized to mitigate the complexity inherent in modern system development, it can be seen that the system engineer must possess several characteristics. A system engineer must be knowledgeable in all phases of the systems engineering process; be able to apply the tools, methods, and methodologies available; and communicate with, direct, and motivate specialists effectively to successfully cope with the issue of complexity.

3.2 FACTORS NECESSITATING SYSTEMS ENGINEERING

In addition to addressing technical complexity, the importance of systems engineering has grown due to factors such as the increasing cost of system acquisition, the often immediate demand for the system, and the recognition of the need for a defined, organized approach to avoid or prevent the mistakes or failures of past system development efforts. [41] These driving forces can be placed into three broad categories:

(1) those resulting from technological advances,

(2) programmatic issues, and

(3) historical context.

3.2.1 Technology Push

The first factor is often referred to as "technology push." [18] It arises from the rapid rate of technological advances creating two basic problems. One problem is the effective implementation of new technologies. Representative issues resulting from rapid technological advances engineers must consider include: How can this new technology be used? Can existing technology support its use or must other technologies be developed prior to the insertion and implementation of the technology of interest? Is the cost of implementing the technology overcome by the benefits to be gained? Stated differently, technology push can be associated with the real or perceived advantage gained by the insertion of high technology.

With the implementation of new technologies, the development of larger, more complex systems often occurs. In addition, introduction of new technologies may increase the time required to perform various system development phases, the requirements for life cycle support, and maintenance and training costs. Thus implementation of new technology may result in longer system development and implementation times with higher acquisition costs.

Increased acquisition time, along with rapid technological advances, leads to a second technological problem: rapid obsolescence. Is the system, in a sense, obsolete by the time it is operationally deployed? [8, 27] System development under these conditions might be acceptable if it were not for the second driving factor: programmatic issues.

3.2.2 Systems Pull

Programmatic issues are often referred to as "systems pull." [18] These issues encompass the programmatic aspects of managing the process. As such, systems pull is concerned with addressing the growth and complexity of modern systems. Programmatic issues include, but are not limited to, the disciplined, orderly development and assessment of the system; interactions with the user or purchaser; satisfaction of the user's needs and demands; the integration of the interdisciplinary efforts; the identification of problems requiring utilization of new technology; and the making of schedule and cost decisions. An example of systems pull is the common situation of the user requirement or demand for immediate implementation of a particular system or technology collateral with reduced funding for development. Immediate need and the trend of reduced funding are programmatic issues particularly applicable to the military community.

These two "driving" factors (technology push and systems pull) can impose conflicting requirements with respect to development time. Insertion of new technology often increases the system development time by lengthening systems engineering development phases in order to ensure the development of effective, safe systems satisfying the user's needs. Conversely, systems pull attempts to effectively apply the systems engineering process to develop systems or implement new technologies at reduced development time and cost. Thus, technology push tends to increase acquisition time while systems pull attempts to reduce it. While systems pull attempts to reduce system development time and cost, it may in fact increase these by identifying new technologies for insertion or by poor implementation of the process.

3.2.3 System Development History Data Base

The third factor may ultimately provide clues for a balance between systems pull and technology push: the history and current science of systems development. As the discipline of systems engineering develops and grows, the assimilated history of lessons learned can be organized, documented, analyzed, and reincorporated back into the systems engineering process. Use of this information can aid in the reduction of development time and cost by avoiding past mistakes or failures. [12] This history base may also provide a means for relating problems similar in nature and facilitate the recognition and identification of broader problem areas within system development activities. Additionally, the solutions or approaches used to resolve or alleviate these problem areas is then available to other system developers confronting analogous problems.

A good example of the interrelationship between the three driving factors can be seen within the system acquisition process of the U.S. Navy. The principal focus of systems engineering literature is on system development from "scratch" in order to satisfy a new need or to replace outdated or inadequate systems. This represents the ideal situation for employing the systems engineering process. However, this is not often truly indicative of the problem faced by most Naval system engineers today: the expansion, modification, or integration of existing Naval combat, surveillance, logistic, navigation, and C^3 systems, or platforms and facilities.

Due primarily to the high system acquisition costs and a current trend of reduced funding, the Navy cannot continually develop and purchase new systems for the hundreds of platforms in today's force or the planned future force. To deploy one system alone on a hundred ships may cost upwards of $100 million. Thus, the problem facing the Navy is not how to develop a new system from ground zero. The problem is how to upgrade, modify, and/or integrate existing systems into larger, synergistic systems resolving or alleviating the current deficiencies. In overcoming this situation, the Naval systems have often evolved in a piecemeal fashion. This piecemeal development often creates new, extremely complex, and unforeseen problems for the system engineers.

The presence of the first two driving factors can easily be seen within the Naval acquisition process. While system obsolescence resulting from new technologies development for the U.S. Navy is a concern, obsolescence caused by the technological advances of potentially hostile nations can be an issue of greater concern. Current Naval system capabilities may be rendered inadequate or obsolete by technological advances such as the reduction of platform and weapon radar cross sectional area or increased weapon performance capabilities in range, speed, and accuracy. The system of interest may be the best available, but it is now inadequate for its intended mission and therefore, in a sense, obsolete.

The enhanced threat capability (technology push) making current Naval systems obsolete is unacceptable to the national defense interests and requires immediate resolution. Programmatically, this may lead to the recognition of the need to implement new technologies (systems pull) to resolve or alleviate the deficiency in system performance against the enhanced threat. As a result of implementing new technology, system complexity may increase.

Looking at the previous example with respect to the role of the third category (the history and current science of systems development), system engineers need to examine systems engineering history to learn from previous efforts. What led to the current situation and what design approaches were utilized under similar system development efforts or situations in the past? Which approaches worked and which failed? How might they be improved upon? What were the system requirements? Were the requirements clearly defined and understood? Were the requirements satisfied?

As the Navy addresses these and other issues, the lessons learned should be incorporated into a systems engineering history base. This history base could be used in the development of expert engineers and scientists, and to aid in future development of new methodologies or modification of existing methodologies that avoid the previous problems encountered. For example, with the historical trend in Naval system development and procurement being one of system evolution, one possible lesson learned may be the requirement for Naval system design to promote system objectives of adaptability and compatibility.

4.0 THE NEED: A MEANS TO DEVELOP AND EVALUATE METHODOLOGIES

The problem being considered of how to improve upon applying the systems engineering process is not the question of the lack of, or the inadequacy of, present design methods and tools. [48] Rather the problem, composed of two parts, is "How to develop and evaluate tools, methods, or methodologies for systems engineering?" and "How to establish understanding of, support of, and communication within the systems engineering process?" These questions need not only to be asked during all phases in the system development of a new system (Figure 1) but also for the situation where modifications or extensions to existing systems are being made.

The following sections introduce and define the concept of objectives, principles, and attributes (OPAs) as it pertains to systems engineering and the above problem. In order to best understand the OPA concept, background information on the software engineering process and the events initiating the development of the concept is provided in sections 4.1 and 4.2.

4.1 THE SOFTWARE ENGINEERING PROCESS

The development of software engineering as a distinct discipline has followed an evolutionary path beginning with the recognition of beneficial techniques for developing the required, reliable software. [1, 21] Its purpose is to provide a consistent, life-cycle approach to the creation of software. [45] By employing the software engineering process, the software requirements required to create software that is economical, reliable, maintainable, and correct are generated through the use of software development methodologies, collection of tools, techniques, and guidelines. [1]

Within the software engineering community several life-cycle models are used to represent the software engineering process. The classic waterfall model [5, 9, 17, 50], the spiral model [5], the Lehman-Stenning-Turski "Two-legged" model [5], and the meta-model [9, 50] represent a few. Each model has merits and limitations in representing the process.

Dandekar [16] presents a classic waterfall software life-cycle model approach consisting of six phases (Figure 3). He proposes all software passes through each of these phases to some extent during its development. The life-cycle phases represent categories of activities but do not necessarily imply a set sequence of occurrence. No one phase is independent of the others; they may occur concurrently and affect each other. Definitions of each of the software engineering phases are provided in the Appendix A.

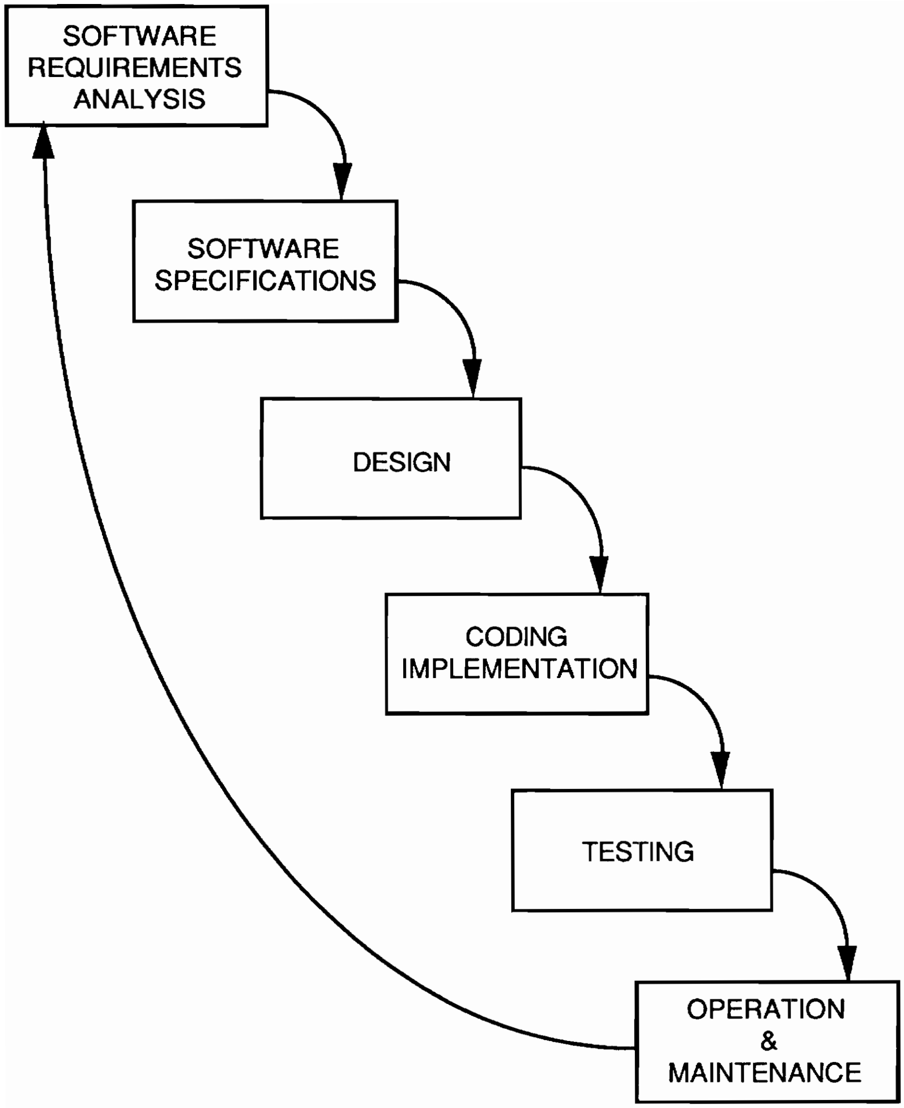

Figure 3. The Software Life Cycle

4.2 THE DEVELOPMENT OF THE SOFTWARE ENGINEERING OBJECTIVES, PRINCIPLES, AND ATTRIBUTES

As the software embedded within systems became more complex, the need for an orderly disciplined approach for software development became crucial. The various software tools available and the methodologies to implement the tools and techniques are an indication of this.

Arthur et al. recognized the lack of studies evaluating available methodologies or the process of developing new methodologies. In response to this need, they developed the concept of objectives, principles, and attributes (OPAs) to aid in understanding, evaluating, developing, or selecting methodologies. [1]

4.3 OBJECTIVES, PRINCIPLES, AND ATTRIBUTES: A BASIS FOR DEVELOPMENT AND EVALUATION

Central to the concept of OPAs and the linkages relating them is the proposal that all methodologies emphasize at least one objective to be obtained by applying that particular methodology. Within each methodology, fundamental design principles guide the engineer in the achievement of the stated system objectives. Thus, the linkage between objectives and principles is that the use of a particular principle or set of principles within a method promotes attainment of a specific objective or set of objectives. [1, 16]

The use of particular principles also induces specific attributes in the final product. Attributes are desirable and beneficial characteristics present at the product level. [1, 16] The attributes can then be used as an indicator to verify the achievement of the desired objective.

It is this relationship among objectives, principles, and attributes that serves not only as the basis for assessing the adequacy of a development methodology, but also as a basis for developing new methodologies. It is recognized that this philosophy is tempered by several practical concerns [1]:

1) While a set of common objectives can be identified, this set might not be complete and additions should be allowed.
2) Objectives can be given different emphasis within a methodology or in the applications of a methodology.
3) Attributes of a large software product might be evident in one component yet missing in others.
4) The evaluation procedure should allow application to varying levels of detail, consistent with the perceived importance of the decisions motivating the effort.

Since systems engineering and software engineering share similar life cycles, Table 1, they may also have similar

Table 1. THE LIFE CYCLES

SYSTEMS ENGINEERING	SOFTWARE ENGINEERING
Identify Need/Problem Definition	Requirements Analysis
Planning	Definition Phase
System Studies	Design
Requirements & Specifications	Coding
Design	Testing
Test & Evaluation	Operation & Maintenance
Production	
Operation & Maintenance	

fundamental objectives and principles. This paper proposes a basis for future development, from which systems engineering methodologies can be developed and evaluated to determine their adequacy. This basis is derived from the concept of software engineering objectives, principles, and attributes as a means for methodology development and evaluation within the discipline of software engineering. The OPAs proposed for software engineering [1, 16] are listed in Table 2. As stated previously, objectives are achieved by implementing specific principles. Table 3 details the relationship between software engineering objectives and the principles employed to obtain them. [1] Further definition of the objectives, principles, and attributes for software engineering can be found in Appendix B.

In summary, the objectives, principles, and attributes concept provides a means for both developing and evaluating systems engineering methods and methodologies.

Table 2. SOFTWARE ENGINEERING OBJECTIVES, PRINCIPLES, AND ATTRIBUTES

OBJECTIVES:

1. Maintainability
2. Correctness
3. Reusability
4. Testability
5. Reliability
6. Portability
7. Adaptability

PRINCIPLES:

1. Hierarchical Decomposition
2. Functional Decomposition
3. Information Hiding
4. Stepwise Refinement
5. Structured Programming
6. Documentation
7. Life-Cycle Verification

ATTRIBUTES:

1. Cohesion
2. Coupling
3. Complexity
4. Well-Defined Interface
5. Readability
6. Ease of Change
7. Traceability
8. Visibility of Behavior
9. Early Error Detection

TABLE 3. SOFTWARE ENGINEERING OBJECTIVES AND CORRESPONDING PRINCIPLES

OBJECTIVE	PRINCIPLES
Maintainability	Stepwise Refinement Documentation Hierarchical Decomposition Functional Decomposition Information Hiding Structured Programming
Adaptability	Stepwise Refinement Documentation Hierarchical Decomposition Functional Decomposition Information Hiding Structured Programming
Reusability	Documentation Hierarchical Decomposition Functional Decomposition Information Hiding
Portability	Functional Decomposition Documentation
Testability	Life-Cycle Verification Hierarchical Decomposition Functional Decomposition Information Hiding Stepwise Refinement Structured Programming
Reliability	Hierarchical Decomposition Information Hiding Stepwise Refinement Structured Programming
Correctness	Life-Cycle Verification Hierarchical Decomposition Stepwise Refinement Structured Programming

5.0 THE LITERATURE SURVEY OF SYSTEMS ENGINEERING PROCESSES

This section is provided to aid the reader in understanding the process of systems engineering. It also discusses the literature survey conducted to determine if a consensus exists on the systems engineering process and objectives, principles, and attributes.

5.1 THE SYSTEMS ENGINEERING PROCESS

The systems engineering process (Figure 4) can be viewed as "a logical sequence of scientific, engineering, and management activities and decisions transforming an operational need into a description of system performance and a preferred system configuration." [31] Through the systems engineering process, functional detail and design requirements evolve to achieve a balance among the stated objectives. The systems engineering process is a continuous and iterative process employed to reach cost-effective design solutions and to ensure that the development effort addresses the correct problem. [28]

The process initiates upon identification of a need and ends with system retirement. Thus, the process encompasses the entire life cycle of a system. [4, 18] Generally no two systems are identical with respect to their goals and

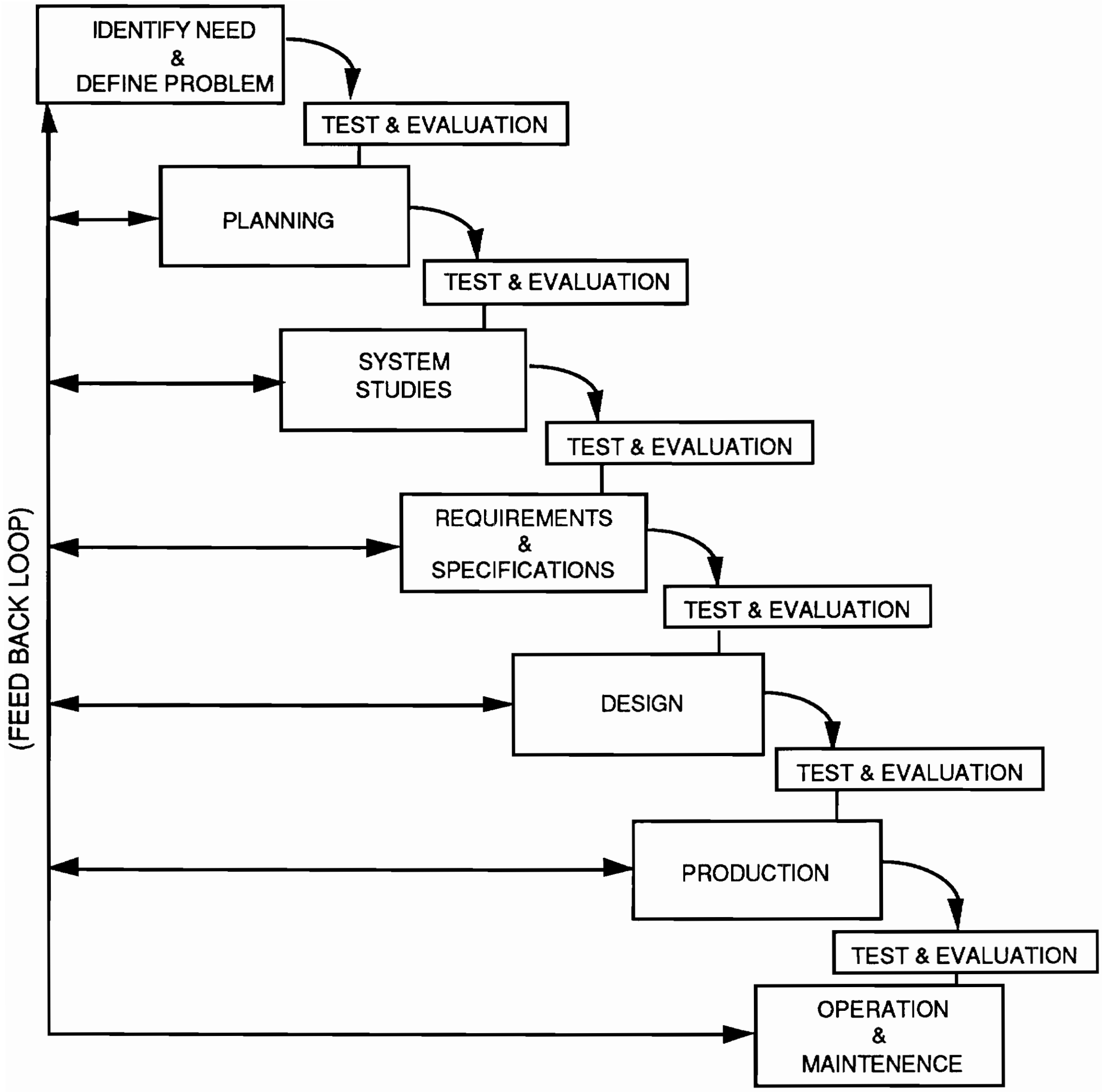

Figure 4. The Systems Engineering Process

objectives; therefore, the process must be tailored specifically to each system being developed. Each phase of the life cycle is considered during the process even though it may not be a major phase in the development, modification, or extension of the particular system of interest. [4]

5.2 THE LITERATURE SURVEY

A survey of systems engineering literature was conducted to determine if a consensus systems engineering process exists. The survey also searched for espoused systems engineering OPAs. The data of the survey are provided in Appendices C, D, and E.

In examining the various systems engineering processes, no readily apparent consensus process exists. Systems engineering processes are hard to compare for several reasons. [36] The proposed phases are often process dependent resulting in wide discrepancies in terminology, definitions, sequence of phases, and activities within the phases. The same phase or activity may appear under different names in different processes. For example, conceptual design in one process may be called architectural design in another. In addition, processes often do not address separate concerns such as problem definition and requirements definition.

The literature often did not explicitly define the phases or the activities performed within a phase. This lack of definition of the phases and activities further complicated the comparison of the processes in the survey effort. However, in examining the processes found within the literature reviewed, several common similarities indicate an identifiable process.

While a consensus was not determined, a representative systems engineering life-cycle process based on these similarities is provided. Table 4 provides a representation of the phases constituting the systems engineering life cycle and example activities occurring within each phase. This is not claimed to be the exclusive systems engineering process or an all-inclusive list of activities to be performed. It is a general representation of the phases evident in the literature surveyed through which a system passes to some extent during its development.

As alluded to previously, a system passes through each phase within the process to some extent during its development. There is no definite sequence of phase occurrence and the order may differ from system to system. Furthermore, it is often difficult to distinguish where one phase ends and another begins since the phases and activities often overlap and interact not only in content of effort but also in time of occurrence.

Table 4. The Systems Engineering Life Cycle

PHASE	ACTIVITIES
Identify Need	Identify the Need & Define Problem Define the System & Objectives Define within Hierarchy of systems
Planning	Program Organization Logistic & Program planning (schedule, milestones, etc.)
System Studies	Exploratory Studies Scientific Research Feasibility & Trade-off Studies Data and Information Collection Analysis (functional, cost, risk, effectiveness, modeling,...)
Requirements & Specifications	Define Systems Constraints Requirements Definition Requirements Synthesis Specification Generation
Design	Architectural Design Functional and/or Hierarchical Decomposition Preliminary & Detailed Design Software & Hardware Design Reliability & Maintainability System Selection System Synthesis & Development Prototype Construction
Test & Evaluation	Demonstration and Validation Requirements Verification Testing and evaluation Data Collection
Production	Construction & Site Preparation Quality Control Distribution Industrial Engineering
Operation	Activation and Logistic Support Monitor Performance Training Maintenance & Operation System Upgrades or Modifications Phase-Out, Recycle, or Disposal

Since the systems engineering process is continuous and iterative, each phase receives feedback from the activities ongoing within other phases (Figure 4). This feedback is utilized to improve previous efforts. Due to the nature of the process being iterative, a particular phase may occur many times within the life of the system. Evaluation of the Test and Evaluation Phase is such an example in that it is usually conducted with each step of development to ensure and verify the achievement of the system goals. In addition, since each system is unique in its objectives and requirements, the process must be "tailored" to each system.

5.2.1 The Systems Engineering Life Cycle

While the life cycles may have differed in the number, naming, and sequence of occurrence of their phases, each basically covered the same areas of development considerations. This becomes even more evident, as well as confusing, when the activities within each stage are considered. An activity conducted within one phase of one process may itself be a phase within another process. Over 200 titles for phases and activities were identified within the survey. The list of phases and activities determined in the literature survey is provided in Appendix C.

The phases of the systems engineering process may in general be defined as follows:

The need identification and problem definition phase initiates the life cycle. It produces a clear, concise statement of the problem or need the system of interest is to be developed to resolve or alleviate. It provides the guiding direction for the subsequent phases.

The planning phase produces such products as project organization, time schedule, milestones, and activities to be performed. In addition, this phase determines or reshapes the next or previous phase and promotes phase initiation under favorable circumstances. [39]

The requirements and specifications phase contains analysis activities resulting in the definition of the external behavior and performance requirements the system must satisfy. Further iterations of this phase produce the detailed specifications for system development. While this phase does not imply how the system will meet its requirements, it may impose design constraints.

The system studies phase is comprised of activities such as exploratory studies, data and information collection, scientific research, critical experiments, etc. This phase includes determining as much history as needed to indicate the events leading to the need or origin of the problem. This phase is often conducted concurrently with the other phases throughout the life cycle.

The design phase is often further broken down to smaller phases such as architecture, preliminary design, detailed design, and final design. Within this phase functional decomposition and/or hierarchical decomposition are performed, and the requirements and specifications are allocated to personnel, equipment, software, or firmware. As design progresses, often the requirements may be found to be inadequate or unachievable. This results in a reiteration of the requirements and specification analysis phase. This phase involves continuous interaction between the system engineer and other involved disciplines.

The test and evaluation phase is conducted throughout the systems engineering process to ensure and verify through testing and program reviews that the end system produced satisfies the system goals, requirements, and specifications. This phase can be accomplished at specified time intervals, during or after other development phases, by testing at various development levels such as component or subsystems, or by combinations of these approaches. In short, test and evaluation occurs throughout the entire life cycle of a system.

The production phase occurs after the system has successfully passed an approved level of design and test and evaluation. The primary objective of the production phase is to produce and deliver an effective, fully supported system. [7]

The operation phase begins with system deployment and ends with system retirement or disposal. Within this phase, activities such as modifications, upgrades, logistic support, and maintenance occur.

Greatest emphasis needs to be applied in the early phases of need identification and problem definition, requirements and specification definition, and design. This emphasis is necessary since these early phases are the points where the greatest impact upon overall system effectiveness and life cycle costs occurs. [43]

5.2.2 Consensus of Systems Engineering Objectives

The majority of sources provided examples of system objectives but stated their objectives were not all-inclusive. The literature recognized that each system to be developed has its own objectives which may or may not be unique. While it is evident there is not one set of objectives appropriate for all systems, there may be a small sub-set of objectives that are common to the majority of systems. This list may be expanded or specific objectives emphasized depending upon the type of systems (military, communication, medical, etc.) to be developed.

For example, the Department of Defense could require all systems to be developed in the future to satisfy the objectives of capability, operability, reliability, maintainability, survivability, and adaptability proposed by Cullen et al. [13] Commercial developers may decide not to consider the objective of survivability.

Seven objectives are proposed as objectives common to the majority of systems to be developed:

(1) Adaptability - the ability of a system to accommodate change. Change may involve adjusting to changes within the environment, modifications to meet new user requirements, or enhancements to improve performance.

(2) Feasibility - operational and economic viability.

(3) Maintainability - the ability of a system or its components to be maintained.

(4) Performance - the functional and operational capability of the system and its components under specified conditions.

(5) Reliability - the probability a system will perform in a satisfactory manner for a specified period of time when operated under specified conditions. [4]

(6) Compatibility - the ability of a system to be integrated with or to support (or be supported by) other systems.

(7) Verifiability - the ability to verify the system satisfies the user's needs and the system requirements.

These objectives are not necessarily independent of one another. For example, the reliability of a system may be decreased if maintainability is neglected within the systems engineering process. As emphasized previously, this list should be tailored to specific industries or categories of systems to be developed.

It is proposed these or another set of objectives be considered for the future development of a Systems Engineering Objectives, Principles, and Attributes approach to system development.

5.2.3 Systems Engineering Principles

The results of the literature survey for systems engineering principles is found in Appendix E. The two principles most commonly referenced within the literature concerning the systems engineering process are hierarchical and functional decomposition. Several sources presented "principles" such as reusability and modularity which may be more appropriately termed as objectives within the context of the terminology of this paper.

Four principles possibly deserving greater consideration within systems engineering are information hiding, life-cycle verification, expansionism, and documentation.

Information hiding is a concept in which the components of the system "know" only the necessary information about the other components comprising the system. System details likely to change are "hidden" inside individual system component design. [34] This is not to say the participants developing the individual components are not aware of the "global" system objectives and the functionality of the other components, but that the system components are unaware of the dynamics ongoing within other components. This principle may be applicable to promoting modular design and thus adaptability within systems. The theory being that modifications or enhancements to one module or subsystem will have minimal impact upon the entire system or other subsystems.

Life-cycle verification verifies the attainment of system requirements throughout the design, development, and maintenance of the life cycle. [16] It also involves verifying that not only are valid principles, tools, and methodologies used throughout the life cycle, but that they are used correctly.

Expansionism involves, and may be defined as, the hierarchy of systems within the realm of systems. Just as it is important to consider the hierarchy of subsystems or functions within a system of interest, it is valuable to consider a system's relationships and position within the hierarchy of systems it interacts with. This principle aids the system engineer in defining the objectives and requirements of the system of interest.

The principle of documentation promotes requirements and decision traceability. This may reduce the redundancy of past and present efforts and ultimately reduce development time and cost.

5.2.4 Systems Engineering Attributes

Attributes are desirable characteristics at the system product level induced by the application of a principle. Attributes are visible and possibly measurable in nature. While Arthur et.al determined software attributes to be generalizable in nature, this may not be the case in systems engineering where the definitive product may be at a different level or scope than that found in software engineering products. As a result, systems engineering attributes may be found to be less formal and more subjective to human judgment in most cases. Systems engineering attributes may also be found to be more measurable in other cases such as mean time between failure. Moreover, the ability to define and measure attributes may also be highly dependent upon the industry applying the process, the model chosen to represent the systems engineering process, and/or the objective(s) of interest.

The literature surveyed contained little discussion of attributes as defined within this paper. In an attempt to provide direction for future development of systems engineering attributes, several examples are provided in Table 5 to illustrate the concept.

Table 5. Systems Engineering Attribute Examples

OBJECTIVE	ATTRIBUTES
Maintainability	Accessibility
	Simplicity
	Mean time between maintenance
	Availability
Reliability	Redundancy
	Mean time between failure
	Availability
Verifiability	Traceability
	Documentation Adequacy

6.0 COMPARISON OF SYSTEMS ENGINEERING TO SOFTWARE ENGINEERING

In order to add validity to the proposal that the concept of OPAs is applicable to systems engineering, it is necessary to establish the relationship and similarities between the disciplines of systems and software engineering. The following sections establish and illustrate these similarities and relationships.

6.1 SIMILARITIES BETWEEN SYSTEMS AND SOFTWARE ENGINEERING

The similarities between systems engineering and software engineering design processes have been well documented. [14, 15, 16, 18, 21, 24, 35, 44] These similarities should not be a surprise if a broad perspective of systems engineering is taken. This perspective is that there are various levels (system, subsystems, components, functions, etc.) of the systems engineering process occurring simultaneously within the system development process. The systems engineering process can be tailored to the scale and scope required and applied at each of these levels by the responsible discipline. Thus, the software engineering process can be viewed as the systems engineering process specifically tailored by software engineers to support the development of software.

6.1.1 The Processes

The similarity of approach and objectives between the two engineering processes is readily apparent. Both processes employ a life-cycle approach to development that initiates with a need and end in retirement. While the objective of software engineering is aimed at creating software that is economical, correct, reliable, and maintainable, the goal of systems engineering is to create systems with similar desirable characteristics. [16] In addition, both processes often employ the same design principles and make similar design decisions and trade-offs. [44] Furthermore, the complexity of systems being developed and the number of participants involved is increasing within both disciplines. [26]

Common process shortcomings are also readily apparent. The lack of or inadequacy of methods, techniques, and tools supporting the entire life cycle often leads to decisions and tradeoffs being based on intuition and experience rather than being supported by a consistent methodology and effective design aids. This is compounded by engineers being familiar with only a few of the implementation methods, techniques, and tools available.[15]

Other common shortcomings include the often inadequate front-end planning; the lack of support or commitment to a total integrated approach to system development; and poor requirements and specifications definition. [43]

6.1.2 The Life Cycles

Having presented representative characterizations of the life cycles in previous sections, a comparison between the life cycles can be made. Not only are the two life cycles similar in appearance, but they are also similar in nature. Both are continuous and iterative processes. While there can be differences between phases of the two life cycles, these in general are minor and may be more related to the nature of the systems being developed than the processes themselves.

Both life cycles initiate with the identification of a need and end with retirement from use. Many phases and activities are common to both disciplines. Problems or difficulties encountered in one phase require a return to previous phases within the life cycle. [44]

Although the phases appear to be precisely defined, the phase and activities within both processes interact and often occur simultaneously. The need identification, planning, and requirements and specification, and system studies phases of systems engineering roughly correlate to the requirements analysis and definition phases of software engineering. Within these phases the need or problem is defined, background studies are conducted, requirements for the system behavior and function are defined, and organization and development plans are made.

The design phase of software engineering correlates to the design phases of systems engineering in that both define the logical details of subsystems, components, and functions. [16] Within these phases the architecture for the system or software code is defined, design constraints determined, and design conducted. It is common for the design phase in both processes to be further subdivided into stages such as preliminary and detailed design.

Test and evaluation of systems engineering corresponds to testing of the software code. Both share the goal of validating and verifying the design and its implementation.

The operation and maintenance phase of systems engineering directly corresponds to operation and maintenance phase of software engineering. Use of the system or software code, logistic support, maintenance, enhancements, and eventually retirement are the primary activities occurring within these phases.

6.2 THE RELATIONSHIP OF SYSTEMS ENGINEERING TO SOFTWARE ENGINEERING

As systems being developed and modified become more software intensive, the need to understand the relationship between systems engineering and software engineering becomes more critical. However, the relationship between the systems engineering and software engineering has not been well documented to date. In a report recommending a training program for software engineers for the U.S. Navy, Warner et al. [49] recognize the lack of distinct definition and delineation of roles and responsibilities between system and software engineers.

This lack of distinction is often viewed as a major stumbling block between the two disciplines. But is it really? The uniqueness of the systems being developed or modified may actually prevent distinctions from being made except on a case-by-case basis. In fact, not delineating between the disciplines may actually be an advantage when considering the relationship between the two disciplines. What is important is to understand the relationship between systems engineering and the disciplines it integrates. The relationship is that the system engineer orchestrates and integrates interdisciplinary efforts within the "overall" systems engineering process. Each discipline conducts its own "tailored" systems engineering process for its assigned task(s), continuously interacting with the other disciplines involved. Thus for example, the software engineering process is the systems engineering process modified toward software development.

Within this relationship, the systems engineering process provides the software engineer and other disciplines with the global requirements for the system, subsystems, or functions allocated to them for development. The software and hardware engineers transform these requirements by employing a "tailored" systems engineering process into more detailed requirements and specifications for the subsystems or functions assigned to them.

As each discipline applies their "tailored" systems engineering process, constant interaction between the system engineer and the disciplines occurs (Figure 5). Exchange of information can occur between any phases of the processes and/or disciplines. Good information exchange between the requirements definition, design, and test and evaluation phases is particularly critical. For example, the software engineer has the responsibility of informing the system engineer of its design considerations which may impact other efforts. The system engineer must integrate and adjudicate these inputs from all disciplines to develop the system as optimally as possible with the available resources.

What is the advantage of this perspective in

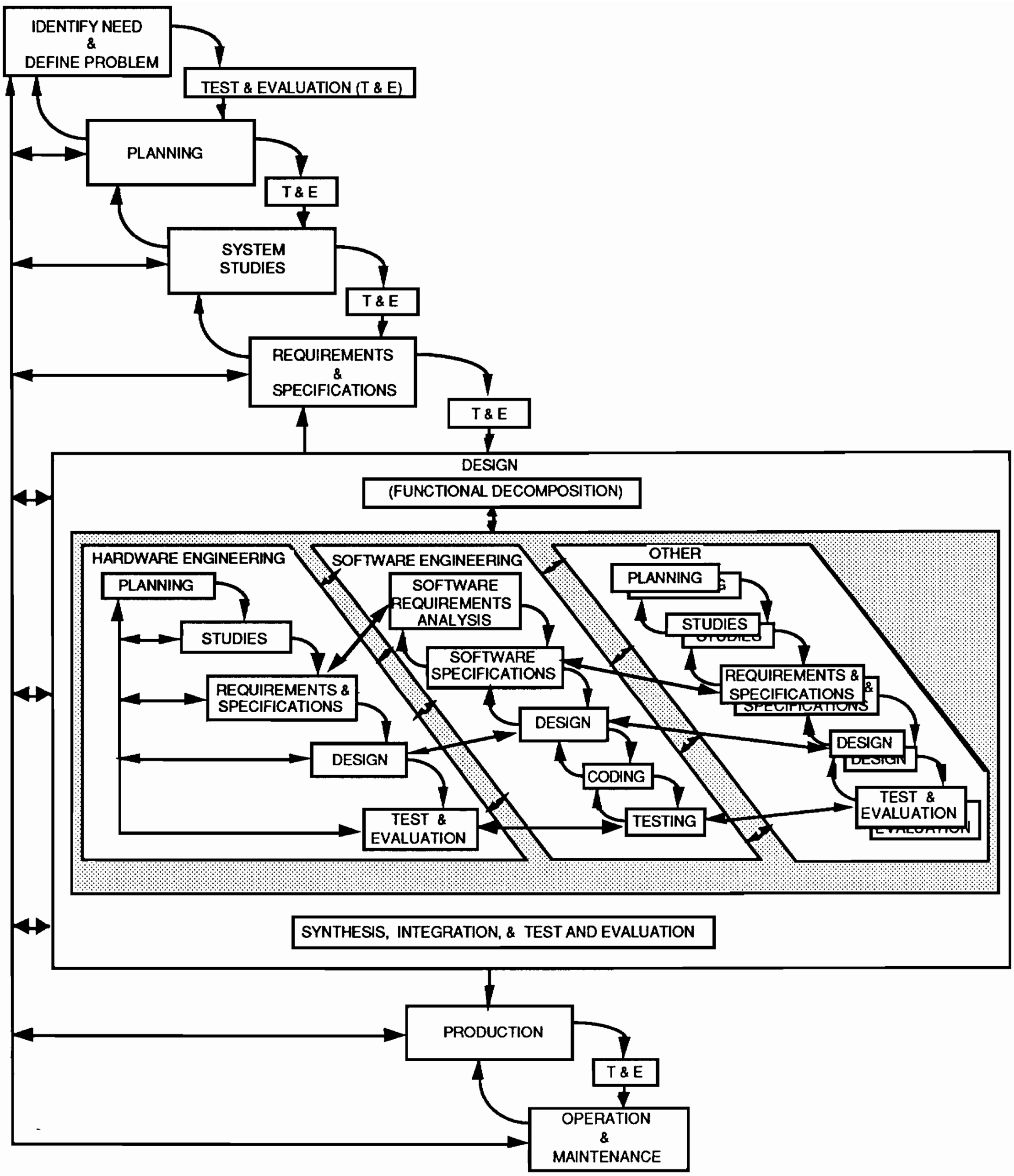

Figure 5. Systems Engineering Process Interactions

understanding the relationship between system and software engineering? It recognizes that the roles and responsibilities of the system and software engineer are similar but of different scope. While of different scope, the roles and responsibilities are dependent, not independent. Each discipline has the responsibility to inform the system engineer of the impacts or requirements its development efforts of subsystems or functions will have upon the "entire" system. This feedback information is incorporated into the iterative systems engineering process. It may be determined that a requirement, or the requirements, must be altered in light of this feedback. Or the feedback may indicate the work of other disciplines involved is directly affected. In this case the previous phases of the systems engineering process are revisited.

An additional advantage is that by not distinctly delineating responsibilities, each discipline must apply the process and thus the phases and activities will not be overlooked. This will improve understanding of the process, enhance communications within the process, and further facilitate the synthesis effort.

7.0 SYSTEMS ENGINEERING POINTS OF VULNERABILITY

While providing a structured approach to system development, the systems engineering process has several vulnerability points that impede the process's implementation. These vulnerability points can be extremely difficult to overcome. No one point is worse than another in that each can be extremely detrimental to the process. These vulnerability points are not "fixed" within the process; they can occur in any phase, within various activities of a phase, at different times, and with different levels of impact. Furthermore, the effects of these vulnerability points can be compounded exponentially by their occurring concurrently. The first step to overcoming these vulnerability points is recognizing and understanding them. The following sections do not claim to provide all the vulnerability points of the systems engineering process. However, these sections do summarize four major vulnerability points.

7.1 UNDERSTANDING THE SYSTEMS ENGINEERING PROCESS

The first vulnerability point is the level of understanding of the systems engineering process by the personnel implementing it. The title of system engineer does not qualify a person as a system engineer. If an engineer does not understand the process or the criticality of the issues, the probability of successfully applying the process is minute. Even understanding the process does not guarantee successful implementation. Crossing the boundary from understanding the process to being able to implement it is often a task so difficult that even the most sincere efforts fail.

7.2 SYSTEMS ENGINEERING IS SUBJECTIVE

A second vulnerability point is that the process is extremely dependent upon human judgment, experience, and intuition. [12, 15, 18, 41] Thus systems engineering often tends to be very subjective. Technology push and system pull discussed in a previous section tend to compound this issue. Engineers may lack the time, financial support, or technical support desired to conduct analyses to eliminate or minimize the subjectivity of the process. Instead, they conduct a level of studies and analysis required to supplement their experience and intuition in making sound engineering decisions.

This does not imply that judgment, intuition, and experience are not valuable. It serves to indicate the need for developing tools and methodologies to aid the system engineer.

7.3 COMMUNICATIONS

Communications is another major vulnerability point. [12, 32, 37] Poor communication of the system's goals and objectives to the specialists (vertical communication) and the communications between specialist areas (horizontal communication) often results in inferior design and synthesis of the subsystems. Poor communications leads to inadequate exchange of the information vital for effective system design and synthesis. In addition, poor communications tend to ripple through and compound the magnitude of the other vulnerability points.

Further harm may be imposed by communicating to the groups only the requirements specific to their subsystem and not also those of the "entire" system. This can be extremely detrimental. Consider for example a design group tasked with developing software code for a navigation system of a ship. If the objectives are not explicitly stated within the requirements, code may be developed to run with extreme accuracy at the cost of speed, reliability, or maintainability. If the true need was for high reliability with less requirement for accuracy, the whole system may be suboptimized. Therefore, the requirements must be identified and understood by all participants.

7.4 PARTICIPANT'S SUPPORT OF THE SYSTEMS ENGINEERING PROCESS

The remaining vulnerability point posed for discussion concerns the level of support of and commitment to the systems engineering process by the project management, system engineers, and participants involved in applying the process of systems engineering. For example, the focus of efforts may emphasize short-term goals of attaining milestones or having "something to show" rather than with the long-term goal of developing the optimum system possible with the resources available. In doing so, economic, political, or social issues are often ignored or overemphasized.

In short, for the process to provide maximum benefit, the systems engineering process must be applied rigorously without cutting corners. The old adage of "You can lead a horse to water but you can't make him drink" is very appropriate. One can emphasize the systems engineering process repeatedly, but the process will not succeed unless all disciplines and participants involved support and are committed to applying the process.

8.0 FUTURE DIRECTIONS FOR INVESTIGATION

To date, the development of the field of systems engineering has primarily been conducted and documented by practicing engineers. [21] Their experience has often been acquired through trial and error. As such, the literature reflects their attempts to try to convey how a system was successfully developed. From this they have formulated general processes which they have successfully employed. These processes provide the system engineer with a mental framework or guideline of the considerations required to increase the probability developing an effective system that solves the right problem.

It is time to go past just providing considerations and a guideline. The engineers need better means of applying the systems engineering process: tools, methods, and methodologies. What is needed to advance beyond this guiding framework is an infrastructure for developing methods, facilitating communications, participant comprehension, and the implementation of the systems engineering process. The concept of objectives, principles,and attributes is one such basis for developing methodologies to implement the process providing an easily understandable means for achieving and communicating the objectives to the interdisciplinary development team.

Three general areas require future investigation, whether conducted separately or in conjunction, to promote further evolution of the field of systems engineering. These areas are the systems engineering environment; tools and methodologies development; and education. Once again, it is proposed that the concept of objectives and principles provides a sound engineering basis to be considered for future developments in each of these areas.

8.1 THE SYSTEMS ENGINEERING ENVIRONMENT

This topic area includes such considerations as the perceived role and responsibilities of systems engineering, and the means for attaining participant support of the process. In considering the systems engineering environment, future developments should consider the following types of paths [14]:

1. Development to minimize the restriction on the use of or the type of tools and/or methodologies. This is to say that the environment should not be so rigid as to preclude the use or development of certain types tools or methodologies. The environment must promote the awareness of the different techniques and methodologies available, and their range of application.

2. The systems engineering environment should provide the means for integrated support throughout the entire life

cycle of system development. Currently tools and methods addressing phases or activities within the system life cycle do not support other ones addressing the same or different phases and activities. In addition, the system engineer may not have a "set" of tools and methodologies that together support development efforts across the entire system life cycle.

3. The systems engineering environment should be developed to be adaptable. Systems engineering addresses complex issues. The needs, requirements, specifications, technology, etc. associated with these issues are constantly changing. The environment must be adaptable to be able to adequately address and respond to these changes.

4. The environment should promote the sharing and utilization of proven concepts across different technologies or between different disciplines.

5. The environment should increase awareness and utilization of methods and techniques for organizing and conducting system development projects with the resources available.

8.2 TOOLS AND METHODOLOGIES DEVELOPMENT

In the area of tools and methodologies, future development should emphasize development of tools, tool environments, and methodologies to support the entire life cycle or particular phases and activities within the life cycle. In particular, those phases occurring early in the life cycle need to be emphasized since it is the early life cycle activities which have the greatest impact on the overall development process. This is due to mistakes or poor decisions made in the early activities being carried and magnified throughout the remainder of the development process. The cost of correcting these can be enormous depending on its magnitude, point of occurrence, time of detection, and delay in responding to the mistakes. Stated differently, it is less costly to do things correctly rather than to have to find and correct flaws that could have been avoided.

At the same time tools are being designed, the methods and methodologies to implement the efficient and effective use of the tools must also be developed. These methodologies should be flexible in the ability to address a wide variety of system development situations. As in tool development, the methodologies should be developed to support not only the entire life cycle, but also each other (that is be integratable).

8.3 TRAINING AND EDUCATION

Education and training is an area the discipline of systems engineering must exploit in the future. A primary goal of systems engineering education should be to develop professionals capable of coping with the diversity of issues inherent in systems development. [39] Systems engineering skills cannot be solely acquired within the classroom. Consequently, programs creating a mixture of academics with hands-on experience is vital. This recognizes that a good system engineer is not developed overnight. Development to become a good system engineer requires years of training and experience.

Educationally, the training of system engineers needs to blend traditional engineering skills with management skills, computer skills, knowledge of the systems engineering process, and an awareness of technology. [3, 19, 39, 40] Additionally, the effects these may have on society must also be instilled. [10, 11, 19, 40]

In coordination with educational programs, development of on-the-job training programs such as co-oping is vital. Programs of this nature can promote the acquisition of experience and confidence in the application of newly learned skills within the context of the systems engineering process.

It is not intended to imply these are the only educational and training areas needing to be addressed.

Other skills also need to be improved or developed, particularly communication skills. Future programs must emphasize the development of oral and written communication skills. [39] Improvement in communication skills can only help facilitate the process.

9.0 CONCLUSION

If the presented definitions of method, methodology, and process are accepted, then systems engineering can be viewed as a process. As such it can be tailored to best stimulate creativity within the development process of each unique system. However, if it is a methodology, creativity may be restricted by the exactness of the methods. This is not to say methods and methodologies should not be utilized by or continued to be developed for use by system engineers. Conversely, it recognizes that each situation is different and that one methodology may not support system engineers across the spectrum of system development applications. Thus, several methods and methodologies may be required to support the entire system development spectrum.

Within this perspective it can been seen that tools, methods, and methodologies are needed to support the involved disciplines in applying the systems engineering process throughout the entire life cycle. Furthermore, these tools and methodologies need to be able to address various levels of detail encountered as the level of detail changes during the iterations of the process.

The objectives, principles, and attribute concept provides an excellent development infrastructure to achieve this goal. The concept of achieving objectives by using specific principles and attributes indicating the attainment of the objective(s) is easily understood by engineers and technicians of all disciplines regardless of their experience level.

The objectives, principles, and attributes concept has two other major benefits. First, the concept may be used to develop, evaluate, or compare existing methods and methodologies for their applicability to specific development situations. The concept overcomes differences in terminology or sequence by focusing on the goals of the methods and methodologies of interest. Thus, by determining which objective(s) are attained by employing a specific method or methodology, the system engineer is provided with a basis for selection of tools which best support the efficient and effective development of the system of interest. The concept aids development by providing the principle(s) for methodologies to be based upon to achieve specific objectives.

The second benefit concerns the area of systems engineering vulnerability points. The ease of understanding and consistent terminology provided by the concept of objectives, principles, and attributes facilitates communications between the inter-disciplinary efforts. This reduces chance of misunderstanding the system's objectives and thus improves design, optimization, synthesis, and integration of the system's components.

The concept of OPAs is an excellent infrastructure for mitigating vulnerability points within the systems engineering process. It is adaptable, flexible, and promotes improved communication and understanding of the systems engineering process by providing consistent terminology. In addition, by providing a means for integrated tool development to support the system engineer, the systems engineering process may become more formalized and less subjective.

The OPA concept is currently just a concept and as such requires further investigation and development. A few areas suggested for future investigation and development are:

- definition of selected objectives and principles with respect to systems engineering,
- definition of their interrelationship,
- definition of desirable systems engineering attributes,
- identification or development of tools and techniques utilizing accepted design principles to achieve specified objectives, and
- development of methodologies based on the concept of objectives, principles, and attributes to integrate these tools and techniques.

10.0 BIBLIOGRAPHY

BIBLIOGRAPHY

1 Arthur, James D., Richard E. Nance, and Sallie M. Henry. "A Procedural Approach to Evaluating Software Development Methodologies: The Foundation", Technical Report SRC-86-008, Systems Research Center, Blacksburg: Virginia Polytechnic Institute and State University, September 1986.

2 Au, Tung, Richard M. Shane, and Lester A. Hoel. Fundamentals of Systems Engineering. Reading, Massachusetts: Addison-Wesley Publishing Co., 1972.

3 Berelter, Susan. "Engineers with a Difference", IEEE Spectrum, Vol. 20, No.1-6, 1983, pp.63-66.

4 Blanchard, Benjamin S., and Wolter J. Fabrycky. Systems Engineering and Analysis. Englewood Cliffs: Prentice-Hall, Inc., 1981.

5 Boehm, Barry W. "A Spiral Model of Software Development and Enhancement," ACM SIGSOFT Software Engineering Notes, VOL II, No. 4, August 1986, pp. 21-42.

6 Boehm, Barry W. "Software Engineering," IEEE Transactions On Computers, VOL. C-25, NO. 12, December 1976, pp. 1226-1240.

7 Booch, Grady. Software Engineering with Ada, Second Edition, Reading, Massachusetts: The Benjamin/Cummings Publishing Company, Inc., 1986.

8 Brooks, Frederick P., Jr. The Mythical Man-Month: Essays on Software Engineering, Reading, Massachusetts: Addison-Wesley Publishing Co., 1975.

9 Callender, E. D. "A Meta-Model for the Software Processes," ACM SIGSOFT Software Engineering Notes, VOL II, No. 4, August 1986, pp. 91-92.

10 Checkland, Peter. "Rethinking A Systems Approach," Journal of Applied Systems Analysis, Vol. 8, 1981, pp. 3-13.

11 Checkland, Peter. "From Optimizing to Learning: A Development of Systems Thinking for the 1990's." Journal of the Operational Research Society, Vol. 36, No. 9, 1985, pp. 757-767.

12 Chestnut, Harold. Systems Engineering Methods. New York: John Wiley & Sons, Inc., 1967.

13 Cullen, Richard P. and Donald L. Hill. Combat System Architecture: Description and Use in Surface Combatants. Dahlgren, VA: Combat Systems Department, Naval Surface Warfare Center, 1981.

14 Cutler, Mel. "An Overview of VLSI Intersected with Software Engineering," IEEE Compcon Digest of Papers, Spring 1983, pp.260-262.

15 Cuykendall, Robert et. al. "Design Synthesis in VLSI and Software Engineering," The Journal of Systems and Software, Vol. 4, No.1, 1984, pp.7-12.

16 Dandekar, Ashok V. A Procedural Approach to the Evaluation of Software Development Methodologies. VPI&SU, 1987.

17 Davis, Alan M. et al. "A Strategy for Comparing Alternative Software Development Life Cycle Models," IEEE Transactions on Software Engineering, Vol. 14, No. 10, OCT 1988.

18 Eisner, Howard. Computer-Aided Systems Engineering. Englewood Cliffs: Prentice-Hall, Inc., 1988.

19 Engineering News, Vol. 36, No.13-18, July-September., 1980, pp.41.

20 Fairley, Richard E. Software Engineering Concepts. New York: McGraw-Hill Book Company, 1985.

21 Hall, Arthur D. "Systems Engineering from an Engineering Viewpoint," Systems Engineering: Methodology & Applications, IEE Press, John Wiley & Sons, Inc., 1977, pp. 13-22.

22 Hall, Arthur D., III. "Three-Dimensional Morphology of Systems Engineering," IEEE Transactions on Systems Science, Vol. SSC-5, pp. 156-160, April 1969.

23 Hill, Douglas J. "Unified Program Planning," Systems Engineering: Methodology & Applications, IEE Press, John Wiley & Sons, Inc.,1977, pp. 23-34.

24 Hirschhorn, S. and A. M. Davis. "Parallels between Software and VLSI Engineering," The Journal of Systems

and Software, Vol. 4, No. 1, 1984, pp. 27-37.

25 Jenkins, Gwilym M. and Philip V. Youle. _Systems Engineering, A Unifying Approach in Industry and Society_. London: C.A. Watts & Co. Ltd., 1971.

26 Katz, Randy, Walt Scacchi, and P. Subrahmanyam. "Environments for VLSI and Software Engineering," _The Journal of Systems and Software_. Vol.4, No.1, 1984, pp. 13-26.

27 Machol, Robert E. _Systems Engineering Handbook_. New York: MacGraw-Hill Book Company, 1965.

28 M'Pherson, P. K. "Systems Engineering: an Approach to Whole-System Design," _The Radio and Electronic Engineer_, Vol. 50, No. 11/12, Nov./Dec. 1980, pp. 545 -558.

29 M'Pherson, P. K. "A Framework for Systems Engineering Design," _The Radio and Electronic Engineer_, Vol. 51, No. 2, February 1981, pp. 59-93.

30 M'Pherson, P. K. "Systems Engineering: a Proposed Definition." _IEEE Proceedings_, Vol. 133, Pt. A, No. 6, Sept. 1986, pp. 330-331.

31 _Military Standard Engineering Management_. MIL-STD-499A(USAF), Department of Defense, 1 May 1974.

32 Nadler, Gerald. "Systems Methodology and Design," _IEEE Transactions on Systems, Man, and Cybernetics_. Vol. SMC-15, No. 6, Nov./Dec. 1985, pp. 685-697.

33 Nance, Richard E. "Data Transfer Architectures: Development of the Capability for Comparative Evaluation," _NSWC-TR 82-345_, Naval Surface Weapons Center, March 1982.

34 Opferman, David C. and Robert H. Yacobellis. "A Design Methodology for System Quality," _AT&T Technical Journal_, Vol. 65 May-June 1986, pp. 60-72.

35 Radar, Jock A. "VLSI and Software Engineering: Guest Editor's Introduction," _The Journal of Systems and Software_. Vol. 4, No.1, 1984, pp. 3-6.

36 Roman, Gruia-Catalin, Mishell J. Stucki, William E. Ball, and Will D. Gillett. " A Total System Design

Framework," Computer, Vol. 7, No. 5, May 1984, pp. 15 -24.

37 Ross, Douglas T., and Kenneth E. Schoman. "Structured Analysis for Requirements Definition", IEEE Transactions on Software Engineering, Vol. SE-3, No.1, January, 1977, pp.6-15.

38 Sage, Andrew P. Methodologies for Large-Scale Systems. New York: McGraw-Hill Book Company, 1977.

39 Sage, Andrew P. "Introduction to Systems Engineering Methodology and Applications," Systems Engineering: Methodology & Applications, IEE Press, John Wiley & Sons, Inc., 1977, pp. 1-10.

40 Seamans, Jr., Robert C., and Kent F. Hansen. "Engineering Education for the Future", Technology Review, Vol 83, February/March, 1981, pp.22-32.

41 Shinners, Stanley M. A Guide to Systems Engineering and Management. Lexington: Lexington Books, D.C. Heath and Company, 1976.

42 Shiners, Stanley M. Techniques of Systems Engineering. New York: McGraw-Hill Book Company, 1967.

43 Singh, Madan G., (Editor). Systems & Control Encyclopedia Theory, Technology, Applications. New York: Pergamon Press, 1988, pp. 995-1003.

44 Smith, Connie U. and John A. Dallen. "A Comparison of Design Strategies for Software and for VLSI," IEEE Compcon Digest of Papers, Spring 1984, pp. 263-268.

45 Systems Engineering Management Guide. Defense Systems Management College, Fort Belvoir, VA., 1986.

46 Truxal, John G., Introductory System Engineering. New York: McGraw-Hill Book Company, 1972.

47 Turner, Wayne C., Joe H. Mize, and Kenneth E. Case. Introduction to Industrial and Systems Engineering. 2nd ed. Englewood Cliffs: Prentice-Hall, Inc., 1987.

48 Warfield, John N. "Selecting Participation Methodologies For Systems Design," 1983 Proceedings of the International Conference on Systems, Man, and Cybernetics, Vol. 2, 1983, pp.762-764.

49 Warner, Walter P., and Richard E. Nance. "The Development of Software Engineers: A View from a User" Technical Report CS 81020-E, Blacksburg: Virginia Polytechnic Institute and State University, 15 Dec. 1981.

50 Wileden, Jack C. "This Is It: A Meta-Model of the Software Process," ACM SIGSOFT Software Engineering Notes, VOL II, No. 4, August 1986, pp. 9-16.

51 Wilson, Brian. Systems: Concepts, Methodologies, and Applications. New York: John Wiley & Sons, 1984.

52 Wilson, Warren E. Concepts of Engineering System Design. New York: McGraw-Hill Book company, 1965.

APPENDIX A

SOFTWARE LIFE CYCLE PHASE DEFINITIONS

SOFTWARE ENGINEERING PHASE DEFINITIONS:

The software development life cycle initiates with the requirements analysis phase. In this phase the need for the software is examined and the requirements are defined. Needs analysis leads to requirements in two forms: design objectives and design constraints.

In the specification phase, software specifications are derived from the defined requirements. These specifications should be directly traceable to the requirements from which they are derived.

The design phase employs the specifications, often in a decomposition approach, to allocate requirements to software components and define the necessary types and forms of component interaction. This phase is typically subdivided into sub-phases such as conceptual design, preliminary design, detailed design, and final design.

In the implementation phase, software coding transforms the final design into executable code. This code must be compilable on the designated computer(s). This phase involves activities such as programming, integration, testing, debugging, and re-design.

The fifth phase is testing the implemented software. This phase ensures the defined specifications are met and satisfied.

The final phase is operation and maintenance.

Maintenance begins with initial operation and continues till retirement of the software. This includes all four forms of maintenance: corrective, adaptive, perfective, and preventive. [16]

Throughout the software life cycle phases, two other critical activities occur. These are verification and validation, and phase reiteration. Verification and validation serves to eliminate as many problems as possible within each phase. Verification involves analysis to ensure that the software product meets its requirements. Validation involves analysis to ensure that the software product satisfies the user's needs. This consists of verifying that not only are valid principles, concepts, and techniques used throughout the life cycle, but that they are used properly. This is necessary to minimize the high cost of correcting poor decisions or design detected late in the life cycle process. Reiteration (or phase wise iteration) recognizes the necessity to repeat phases to rectify problems detected during verification and validation or to make general improvements. [16]

APPENDIX B

SOFTWARE ENGINEERING OBJECTIVES, PRINCIPLES, AND ATTRIBUTES DEFINITIONS

Appendix B provides a brief description of software engineering objectives, principles, and attributes as presented by Dandekar and Arthur. [1, 16]

OBJECTIVES:

1. Maintainability: the ability to make corrections or modifications to the software code in response to recognized inadequacies or deficiencies.
2. Correctness: the strict adherence to the specified software requirements.
3. Reusability: the ability to use software developed for a particular application in other applications.
4. Testability: the ability to evaluate software conformance with the specified requirements.
5. Reliability: the error-free use of software code over a given period of time.
6. Portability: the ease in transferring software to another environment.
7. Adaptability: the ability of a software application to accommodate enhancement or extension modifications.

PRINCIPLES:

1. Hierarchical decomposition: defining software components in a top-down manner from the highest-level of decisions to be made to the lowest level.
2. Functional decomposition: the partitioning of software

components by functional boundaries.

3. Information hiding: insulating the internal details of component behavior so that data or procedures subject to change in one component are "hidden' from the other components.
4. Stepwise refinement: utilizing a convergent design process.
5. Structure programming: using a restricted set of control constructs to result in easier to understand code.
6. Documentation: the management of supporting documents (system specifications, user manuals, etc.) throughout the life cycle.
7. Life cycle verification: verification of requirements throughout the design, development, and maintenance of the life cycle.

ATTRIBUTES:

1. Cohesion: the strength of association of statements within a software component.
2. Coupling: the interdependence among software components.
3. Complexity: an abstract measure of work associated with developing a software component.
4. Well-defined interface: the definitional clarity and completeness of a shared boundary between a pair of software components.
5. Readability: the difficulty in understanding a software

component (related to complexity)

6. Ease of change: the ease with which software enhancements or extensions can be implemented.
7. Traceability: the ease in retracing the complete history of a software component from its current status to its design inception.
8. Visibility of behavior: the provision of a review process for error checking.
9. Early error detection: indication of faults in requirement's specification and design prior to implementation.

APPENDIX C

LITERATURE SURVEY RESULTS ON SYSTEMS ENGINEERING PHASES, ACTIVITIES, AND CHARACTERISTICS

Appendix C provides the survey data on systems engineering phases, activities, and characteristics identified in the literature survey. An "X" indicates an espoused phase within the systems engineering process presented within a given source. Numbers indicated an activity. The number references the phase in which that activity is conducted. An "*" represents an activity within a process common to one or more phases, or a characteristic of the process.

The numbers at the top of each column, under "SURVEY SOURCE", correspond to the bibliography entry with the same number.

		SURVEY SOURCE																					
SYSTEMS ENGINEERING PROCESS PHASES AND ACTIVITIES		1	2	4	6	10	11	12	13	14	15	16	18	21	22	24	25	26	27	28	29	30	31
1	Acquire Facilities & Resources																			2			
2	Acquisition																			X			
3	Action to Solve or Improve Situation																						
4	Activation & Support																			2			
5	Allocation (Functional)												X										X
6	Alternatives																176						
7	Analysis			178																	X		
8	Analysis of Alternatives																						
9	Analyze How Quality Incr. Satisfact.																						
10	Architectural Design											X						X					
11	Architecture																						
12	Assemble & Test																			2			
13	Assess Benefits of New Technology																						
14	Background on Needs & Environ.					61								182									
15	Bldg, Testing, & Acceptance																						
16	Block															X							
17	Circuit															X							
18	Circuit Design Stage																						
19	Communicating Results													58									
20	Comparison																						
21	Conceptual Analysis																						
22	Conceptual Design			176																144			
23	Conceptualization																						
24	Constraints																				135		
25	Construction											X								2			
26	Control																176						
27	Coordination																						
28	Correlating Test Data to Obj.&Req.																						
29	Current Engineering													X									
30	Data Collection			175																			
31	Debate Model Comparitson to Real World					X																	
32	Decision														*								
33	Define Boundaries & Constraints																						
34	Define Desirable Changes					X																	
35	Define System Elements																						
36	Definition of Need																						

		SURVEY SOURCE																					
SYSTEMS ENGINEERING PROCESS PHASES AND ACTIVITIES		1	2	4	6	10	11	12	13	14	15	16	18	21	22	24	25	26	27	28	29	30	31
37	Demonstration/Validation																						
38	Design																			144	116		
39	Design Constraints			176																			
40	Design Refinement																						
41	Detailed Design			176								X						X		144			
42	Determine Purpose within Hierarchy of Purposes																						
43	Develop & Detail Solution																						
44	Develop Feasible Solution																						
45	Development																			144			
46	Development Planning													X									
47	Distribution														X					2			
48	Documentation																179	X					
49	Documentation & Installation																						
50	Effective Analysis Modeling																						123
51	Effectiveness Analysis Simulation																						
52	Environmental Factors																						109
53	Equipment Development							X															
54	Equipment Design							X															
55	Equipment Specifications																			144			
56	Evaluation			150														X	X	144			
57	Evaluation & Optimization																				X		
58	Evolution																	X					
59	Examination After Installation																						
60	Exploratory Planning													X									
61	Exploratory Studies																						
62	Express Problem Situation					X																	
63	Feasibility Study & Prelim. Design			181																			
64	Feedback			*				*		*			*	*						*			
65	Final Design, Construction, & Field Observation																						
66	Finding Out																						
67	Firmware Design Stage																						
68	Follow-up													29									
69	Forecasting																						
70	Formulation																						
71	Formulation of Evaluation Criteria																						

	SYSTEMS ENGINEERING PROCESS PHASES AND ACTIVITIES	SURVEY SOURCE 1	2	4	6	10	11	12	13	14	15	16	18	21	22	24	25	26	27	28	29	30	31
72	Formulation of Problem																						
73	Functional Analysis												X										X
74	Functional Decomposition												72										
75	Functional Requirements																						109
76	Generation of Specifications																						X
77	Generate Alternatives Ideas																						
78	Generation of Specifications												X										
79	Goals & Objectives																						
80	Human Activity (Determine)					X																	
81	Identify Need			X																			
82	Implement Changes					X																	
83	Implementation																	X					
84	Improved Operation																						
85	Initial Design Studies																			144			
86	Initial Operation																180						
87	Initiation																		X				
88	Initiation and Acquisition																	X					
89	Insertion																			2			
90	Install Proposed Design & Follow Up																						
91	Integration																						
92	Interpretation																						
93	Iteration			*		*		*		*		*	*						*	*	*		*
94	Layout															X							
95	Life Cycle Cost Engineering			*																			
96	Life Cycle			*				*		*		*	*	*						*	*		
97	Life Cycle Cost Analysis												X										137
98	Life Cycle Support																						
99	List Inputs,Outputs,&Relationship																						
100	Logic															X							
101	Logic Design											X											
102	Logistic Engineering												X										X
103	Logistic Specifications																			144			
104	Logistic Support Analysis																						98
105	Logistic Support Modeling																						98
106	Machine Design Stage																						
107	Maintenance											X								122			
108	Maintenance Engineering Analysis																						98

	SYSTEMS ENGINEERING PROCESS PHASES AND ACTIVITIES	SURVEY SOURCE																					
		1	2	4	6	10	11	12	13	14	15	16	18	21	22	24	25	26	27	28	29	30	31
109	Management																						
110	Manufacturing Engineering																			2			
111	Marketing & Deployment																			122			
112	Marketing & Sales																			122			
113	Mission & Requirements Analysis																						X
114	Mission Analysis																						
115	Mission Objectives																						109
116	Mission Requirements Analysis												X										
117	Model Building (Build Models)			176		X											176						
118	Modelling																				X		
119	Modifications			201																			
120	Modify																			148			
121	Monitor Performance																			122			
122	Needs Statement																				135		
123	Objectives									*				58			189				135		
124	Operation			201											X					X			
125	Operation & Maintenance																						
126	Operational Characteristics																						109
127	Optimization												X										X
128	Optimization of Alternatives														*								
129	Organization (of Project)																		X				
130	Packaging																			2			
131	Performance Criteria																189						
132	Performance Requirements																						71
133	Phase-out, Disposal, Reclamation, & Recycle			X																154			
134	Planning			181																144			
135	Planning for Action														*								
136	Preliminary Design			176															X	144			
137	Preliminary synthesis																						
138	Principal Design																		X				
139	Problem Definition					61								58	*		189				X		
140	Producibility Analysis																						137
141	Production														X					2			
142	Production & Quality Control																			2			
143	Production Engineering Analysis			140									X										X
144	Production/Construction			X																			

	SURVEY SOURCE																						
SYSTEMS ENGINEERING PROCESS PHASES AND ACTIVITIES		1	2	4	6	10	11	12	13	14	15	16	18	21	22	24	25	26	27	28	29	30	31
145	Program Planning													182	X								
146	Project Planning														X								
147	Prototype Construction			176															X				
148	Prototype Construction & Evalution																						
149	Realization																			X			
150	Realization Decomposition																						
151	Recruit & Train Personnel																			2			
152	Reliability																176						
153	Renovation																			X			
154	Repair Level Analysis																						98
155	Requirements													164		X					135		
156	Requirements Analysis																	X					
157	Requirements Definition											X											
158	Requirements Synthesis																						
159	Retirement														X					X			
160	Root Definitions (Develop)					X																	
161	Sanction																179			144			
162	Search																						
163	Selecting Optimum System													58									
164	Selection																				X		
165	Selection and Partition																	X					
166	Sell or Scrap																			154			
167	Simulation																						
168	Site Preparation																			2			
169	Software Design Stage																						
170	Specifications																						
171	Statement of Needs & Objectives																						
172	Studies During Development													X									
173	Support																			122			
174	Symbolic															X							
175	Synthesis												X				176				X		X
176	Synthesis of Alternatives																						
177	System Architecture Design																						
178	System Concept							X															
179	System Decomposition																						
180	System Definition																189						
181	System Design Function			X				X									X						

		SURVEY SOURCE																					
SYSTEMS ENGINEERING PROCESS PHASES AND ACTIVITIES		1	2	4	6	10	11	12	13	14	15	16	18	21	22	24	25	26	27	28	29	30	31
182	System Development							X							X								
183	System Evaluation Function			X				X															
184	System Implementation																X						
185	System Operation																X						
186	System Planning Function			X																			
187	System Requirements																						
188	System Research Function			X																			
189	System Retirement																						
190	System Specifications																	X		144			
191	System Studies			183										X									
192	System Synthesis													58	*					144			
193	System Upgrades																						
194	System Utilization																						
195	System/Cost Effectiveness Analysis																						123
196	Systems Analysis													58	*		X			144			
197	Systems Analysis & Modeling																						
198	Testing																	X	X				
199	Testing & Integration											X											
200	Trade-off Studies																						123
201	Training																		X				
202	Understand the Competition																						
203	Understand the User																						
204	Understand the User's Environment																						
205	Update Specifications																						
206	Updating					34																	
207	Updating Equipment Charcteristics																						
208	Use																	X					
209	Use & Logistic Support			X																			
210	Validation																						
211	Value System Design														*								
212	Verification															X							
213	Wider system Definition																189			*			
214	Wider System Objectives																189			*			
215																							

		SURVEY SOURCE																				
SYSTEMS ENGINEERING PROCESS PHASES AND ACTIVITIES		32	33	34	35	36	37	38	39	41	42	43	44	45	46	48	49	51C	51H	51J	51R	52
1	Acquire Facilities & Resources																					
2	Acquisition																					
3	Action to Solve or Improve Situation																	X				
4	Activation & Support																					
5	Allocation (Functional)																					
6	Alternatives								186										183		155	
7	Analysis																					
8	Analysis of Alternatives													113								
9	Analyze How Quality Incr. Satisfact.			X																		
10	Architectural Design				X																	
11	Architecture												X									
12	Assemble & Test																					
13	Assess Benefits of New Technology			X																		
14	Background on Needs & Environ.																					
15	Bldg, Testing, & Acceptance																					
16	Block																					
17	Circuit																					
18	Circuit Design Stage					X																
19	Communicating Results																					
20	Comparison																	X				
21	Conceptual Analysis									X												
22	Conceptual Design											X										
23	Conceptualization					135								X								
24	Constraints								135	21												
25	Construction				X															81		
26	Control																			173		
27	Coordination																					
28	Correlating Test Data to Obj.&Req.									X												
29	Current Engineering																					
30	Data Collection																			155		
31	Debate Model Comparitson to Real World																					
32	Decision							*	X		*											
33	Define Boundaries & Constraints													114							68	
34	Define Desirable Changes																	X				
35	Define System Elements													*								
36	Definition of Need											X										

		SURVEY SOURCE																				
SYSTEMS ENGINEERING PROCESS PHASES AND ACTIVITIES		32	33	34	35	36	37	38	39	41	42	43	44	45	46	48	49	51C	51H	51J	51R	52
37	Demonstration/Validation													X								
38	Design																					
39	Design Constraints					135																
40	Design Refinement									28												
41	Detailed Design				X							X		45								
42	Determine Purpose within Hierarchy of Purposes	X																				
43	Develop & Detail Solution	X																				
44	Develop Feasible Solution	X																				
45	Development												142	X								X
46	Development Planning																					
47	Distribution										X											
48	Documentation																		156	81		
49	Documentation & Installation																					
50	Effective Analysis Modeling																					
51	Effectiveness Analysis Simulation									X												
52	Environmental Factors																					
53	Equipment Development																					
54	Equipment Design																					
55	Equipment Specifications																					
56	Evaluation					*													156		X	
57	Evaluation & Optimization																					
58	Evolution																					
59	Examination After Installation																					
60	Exploratory Planning																					
61	Exploratory Studies					*																
62	Express Problem Situation																					
63	Feasibility Study & Prelim. Design									21												X
64	Feedback									*												
65	Final Design, Construction, & Field Observation																					X
66	Finding Out																	X				
67	Firmware Design Stage					X																
68	Follow-up																					
69	Forecasting																			173		
70	Formulation																				X	
71	Formulation of Evaluation Criteria																					

	SURVEY SOURCE																				
SYSTEMS ENGINEERING PROCESS PHASES AND ACTIVITIES		32	33	34	35	36	37	38	39	41	42	43	44	45	46	48	49	51C	51H	51J	51R	52
72	Formulation of Problem																			136		
73	Functional Analysis									X		22										
74	Functional Decomposition												81									
75	Functional Requirements																					
76	Generation of Specifications																					
77	Generate Alternatives Ideas	X																				
78	Generation of Specifications																					
79	Goals & Objectives																					
80	Human Activity (Determine)																					
81	Identify Need								135													
82	Implement Changes																					
83	Implementation												X							X		
84	Improved Operation																			120		
85	Initial Design Studies																					
86	Initial Operation																			120		
87	Initiation																					
88	Initiation and Acquisition																					
89	Insertion																					
90	Install Proposed Design & Follow Up	X						X														
91	Integration					*																
92	Interpretation																				X	
93	Iteration					*				*		*										
94	Layout																					
95	Life Cycle Cost Engineering																					
96	Life Cycle											X										
97	Life Cycle Cost Analysis																					
98	Life Cycle Support											206										
99	List Inputs,Outputs,&Relationship																		133			
100	Logic																					
101	Logic Design				X																	
102	Logistic Engineering																					
103	Logistic Specifications																					
104	Logistic Support Analysis																					
105	Logistic Support Modeling																					
106	Machine Design Stage					X																
107	Maintenance																					
108	Maintenance Engineering Analysis																					

		SURVEY SOURCE																				
SYSTEMS ENGINEERING PROCESS PHASES AND ACTIVITIES		3 2	3 3	3 4	3 5	3 6	3 7	3 8	3 9	4 1	4 2	4 3	4 4	4 5	4 6	4 8	4 9	51C	51H	51J	51R	5 2
109	Management																					
110	Manufacturing Engineering																					
111	Marketing & Deployment																					
112	Marketing & Sales																					
113	Mission & Requirements Analysis									X												
114	Mission Analysis													X								
115	Mission Objectives																					
116	Mission Requirements Analysis																					
117	Model Building (Build Models)																	X		173	55	
118	Modelling																					
119	Modifications											140		124								
120	Modify																					
121	Monitor Performance																					
122	Needs Statement																					
123	Objectives								209	X									X	136	68	
124	Operation							X			X			X						X		
125	Operation & Maintenance				X																	
126	Operational Characteristics																					
127	Optimization							*				22										
128	Optimization of Alternatives										*											
129	Organization (of Project)																			186		
130	Packaging																					
131	Performance Criteria									109												
132	Performance Requirements																					
133	Phase-out, Disposal, Reclamation, & Recycle																					
134	Planning																					
135	Planning for Action							*	X		*								X			
136	Preliminary Design																					
137	Preliminary synthesis											22										
138	Principal Design																					
139	Problem Definition					X		*	X		*							64	X		68	
140	Producibility Analysis																					
141	Production							X			X			X								
142	Production & Quality Control																					
143	Production Engineering Analysis																					
144	Production/Construction											X										

	SURVEY SOURCE																				
	SYSTEMS ENGINEERING PROCESS PHASES AND ACTIVITIES	32	33	34	35	36	37	38	39	41	42	43	44	45	46	48	49	51C	51H	51J	51R	52
145	Program Planning							X			X											
146	Project Planning							X			X											
147	Prototype Construction																					
148	Prototype Construction & Evalution											40										X
149	Realization												X									
150	Realization Decomposition																					
151	Recruit & Train Personnel																					
152	Reliability																			173		
153	Renovation																					
154	Repair Level Analysis																					
155	Requirements				X					109												
156	Requirements Analysis									X												
157	Requirements Definition												11									
158	Requirements Synthesis																					
159	Retirement							X			X											
160	Root Definitions (Develop)																	157				
161	Sanction																			81		
162	Search																				X	
163	Selecting Optimum System									X									X			
164	Selection					*												X				
165	Selection and Partition																					
166	Sell or Scrap																					
167	Simulation									21										173		
168	Site Preparation																					
169	Software Design Stage					X																
170	Specifications																					
171	Statement of Needs & Objectives																					X
172	Studies During Development																					
173	Support													124								
174	Symbolic																					
175	Synthesis																					
176	Synthesis of Alternatives																					
177	System Architecture Design					176																
178	System Concept																					
179	System Decomposition												11									
180	System Definition											22								186		
181	System Design Function					X						40								X		

SURVEY SOURCE																						
SYSTEMS ENGINEERING PROCESS PHASES AND ACTIVITIES		32	33	34	35	36	37	38	39	41	42	43	44	45	46	48	49	51C	51H	51J	51R	52
182	System Development							X			X											
183	System Evaluation Function																					
184	System Implementation																					
185	System Operation																					
186	System Planning Function																					
187	System Requirements					135																
188	System Research Function																					
189	System Retirement											X										
190	System Specifications												11									
191	System Studies																					
192	System Synthesis							*	X	X	*	22							X			
193	System Upgrades																					
194	System Utilization											X										
195	System/Cost Effectiveness Analysis																					
196	Systems Analysis							*			*								X	X		
197	Systems Analysis & Modeling								X													
198	Testing									X				45							199	
199	Testing & Integration				X																	
200	Trade-off Studies									X												
201	Training																					
202	Understand the Competition			X																		
203	Understand the User			X																		
204	Understand the User's Environment			X																		
205	Update Specifications																					
206	Updating																					
207	Updating Equipment Charcteristics									X												
208	Use																					
209	Use & Logistic Support																					
210	Validation					*																
211	Value System Design							*	X		*											
212	Verification					*															X	
213	Wider system Definition																			186		
214	Wider System Objectives																			186		
215																						

APPENDIX D

SYSTEMS ENGINEERING OBJECTIVES

		SURVEY SOURCE																					
SYSTEMS ENGINEERING OBJECTIVES		1	2	4	6	10	11	12	13	14	15	16	18	21	22	24	25	26	27	28	29	30	31
1	Adaptabiliy								X														
2	Applicability																						
3	Availability																			X	X		
4	Capability								X												X		
5	Compatibility							X												X			X
6	Complexity																						
7	Configuration control												X										
8	Consistency											X											
9	Cost							X					X					X		X	X		
10	Dependability																						
11	Design Correctness											X											
12	Design Description																						
13	Design Quality																						
14	Economic Feasibility																						
15	Efficiency											X											
16	Environmental Compatibility											X											
17	Equivalency																						
18	Feasibility											X											
19	Flexibility											X											
20	Human Factors																						
21	Implementability											X											
22	Interoperability												X										
23	Life Expectancy																						
24	Maintainability			X				X	X				X			X				X	X	X	
25	Manability			X																			
26	Marketability																			X			
27	Modifiability											X											
28	Operability								X											X			
29	Performance			X				X		X	X	X	X					X			X	X	
30	Power Consumption																						
31	Producibility			X									X										
32	Quality																						
33	Readiness																						
34	Reliability			X				X	X	X	X	X	X			X	X		X	X	X	X	
35	Reuseability									X		X						X					
36	Robustness												X										
37	Safety																			X			

		SURVEY SOURCE																					
SYSTEMS ENGINEERING OBJECTIVES		1	2	4	6	10	11	12	13	14	15	16	18	21	22	24	25	26	27	28	29	30	31
38	Schedule																						
39	Security										X												
40	Speed										X												
41	Standardization												X										
42	Supportability			X																X	X	X	
43	Survivability			X					X											X			
44	Testability									X		X				X		X					
45	Throughput																						
46	Time							X															
47	Traceability												X										X
48	Transmission Accuracy																						
49	Transportability																						
50	Understandability											X											
51	Useability																						
52	Weight																						

		SURVEY SOURCE																				
SYSTEMS ENGINEERING OBJECTIVES		32	33	34	35	36	37	38	39	41	42	43	44	45	46	48	49	51C	51H	51J	51R	52
1	Adaptabiliy																					
2	Applicability		X																			
3	Availability		X	X																		
4	Capability																					
5	Compatibility			X																		
6	Complexity												X									
7	Configuration control																					
8	Consistency						X						X									
9	Cost		X	X						X		X		X								
10	Dependability											X										
11	Design Correctness												X									
12	Design Description												X									
13	Design Quality																					
14	Economic Feasibility																					
15	Efficiency												X									
16	Environmental Compatibility																					
17	Equivalency												X									
18	Feasibility						X						X									
19	Flexibility		X				X						X									
20	Human Factors											X										
21	Implementability												X									
22	Interoperability																					
23	Life Expectancy									X												
24	Maintainability		X			X				X		X		X								
25	Manability																					
26	Marketability																					
27	Modifiability												X									
28	Operability																					
29	Performance									X		X	X	X								X
30	Power Consumption									X												
31	Producibility											X		X								
32	Quality			X									X									
33	Readiness																					
34	Reliability			X		X				X		X	X	X								
35	Reuseability																					
36	Robustness		X																			
37	Safety													X								

		SURVEY SOURCE																				
SYSTEMS ENGINEERING OBJECTIVES		**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**	**41**	**42**	**43**	**44**	**45**	**46**	**48**	**49**	**51C**	**51H**	**51J**	**51R**	**52**
38	Schedule									X				X								
39	Security																					
40	Speed																					
41	Standardization																					
42	Supportability											X		X								
43	Survivability						X					X		X								
44	Testability						X						X									
45	Throughput																					
46	Time																					
47	Traceability						X							X								
48	Transmission Accuracy		X																			
49	Transportability													X								
50	Understandability												X									
51	Useability			X										X								
52	Weight									X												

APPENDIX E

SYSTEMS ENGINEERING PRINCIPLES

		SURVEY SOURCE																					
SYSTEMS ENGINEERING PRINCIPLES		1	2	4	6	10	11	12	13	14	15	16	18	21	22	24	25	26	27	28	29	30	31
1	Abstraction										X	X											
2	Completeness																						
3	Confirmability											X											
4	Functional Decomposition			X	*			X	X			X	X			X					X		
5	Hiding											X											
6	Hierachial Decomposition					X		X	X			X	X			X	X				X		
7	Localization (Hierarchical)											X											
8	Modularity																						
9	Regularity																						
10	Reusability									X													
11	Simplicity																						
12	Structured Design											X											
13	Symmetry																						
14	Uniformity											X											
15																							

SURVEY SOURCE																						
SYSTEMS ENGINEERING PRINCIPLES		32	33	34	35	36	37	38	39	41	42	43	44	45	46	48	49	51C	51H	51J	51R	52
1	Abstraction												X									
2	Completeness												X									
3	Confirmability												X									
4	Functional Decomposition	X								X		X		X								X
5	Hiding			X									X									
6	Hierachial Decomposition	X					X	X		X			X	X						X		
7	Localization (Hierarchical)												X									
8	Modularity						X															
9	Regularity												X									
10	Reusability																					
11	Simplicity												X									
12	Structured Design																					
13	Symmetry												X									
14	Uniformity												X									
15																						

VITA

David K. Kreider is an electronics engineer for the Warfare Systems Architecture and Engineering Program Office of the Naval Surface Warfare Center, Dahlgren, VA. His current primary responsibilities are to support the assessment of Naval Forces performance, to support the architectural determination for future Naval evolution, and to conduct Battle Force systems engineering to implement the architecture.

David Kreider, born 5 April 1961, received a B.S. degree in Engineering Science and Mechanics from the University of Tennessee at Knoxville, in 1984.

David K. Kreider