AD-A277 702

NSWCDD/TR-92/268

# SYSTEMS DESIGN FACTORS:
## *The Essential Ingredients of System Design*
## Version 0.4

BY CUONG M. NGUYEN AND STEVEN L. HOWELL

ADVANCED SYSTEMS TECHNOLOGY GROUP
SYSTEMS RESEARCH AND TECHNOLOGY DEPARTMENT

18 MARCH 1994



NAVAL SURFACE WARFARE CENTER
DAHLGREN DIVISION • WHITE OAK DETACHMENT
Silver Spring, Maryland 20903-5640

NSWCDD/TR-92/268

# SYSTEMS DESIGN FACTORS:

## *The Essential Ingredients of System Design*

## Version 0.4

BY CUONG M. NGUYEN AND STEVEN L. HOWELL
ADVANCED SYSTEMS TECHNOLOGY GROUP
SYSTEMS RESEARCH AND TECHNOLOGY DEPARTMENT

18 MARCH 1994

| Accesion For | |
|---|---|
| NTIS CRA&I | X |
| DTIC TAB | |
| Unannounced | |
| Justification | |
| By | |
| Distribution / | |
| Availability Codes | |
| Dist | Avail and / or Special |
| A-1 | |

Approved for public release; distribution is unlimited.

NAVAL SURFACE WARFARE CENTER
DAHLGREN DIVISION • WHITE OAK DETACHMENT
Silver Spring, Maryland 20903-5640

## FOREWORD

The results presented in this draft are part of the Office of Naval Technology (Code ONT-227) Engineering of Complex Systems (ECS) Technology Block effort. The ECS block was developed to integrate systems engineering capabilities for developing large scale, real-time, computer intensive systems. The goal of the block is to improve the way in which the Navy currently creates, maintains, and improves systems by incorporating state-of-the-art technology and supplying new technology where holes in present methods exist. The block is divided into four projects: Systems Design Synthesis Technology (RS34P11), Systems Evaluation and Assessment Technology (RS34P12), Systems Reengineering Technology (RS34P13), and Engineering Application Prototype (RS34P14). These projects work closely together to incorporate new technology across the entire system development life cycle.

The System Design Factors development is a collaborative effort among the tasks within the System Design Synthesis Project. This effort is coordinated by the System Design and Structuring and Allocation Optimization Task. The goal of this effort is to generate a list of System Design Factors which are intended for use throughout the whole system engineering process. For instance, they are used to specify in the requirements phase, encapsulate in the capturing phase, quantify and evaluate in the analysis phase, characterize in the optimization phase, justify in the design trade-off phase. These factors are critical to the system engineering process.

The lessons learned in this effort will benefit the whole systems engineering community. The list is expected to evolve as this effort progresses. This is a collaborative effort among Naval Surface Warfare Center Dahlgren Division (NSWCDD), DoD, other government agencies, industry, and university communities.

The authors would like to thank Ngocdung T. Hoang, My-Hanh N. Trinh, Charles Whelan, Jr., Eric Ogata, Dong Choi, Micheal Jenkins, Micheal Edwards, Dr. Ed Cohen, Dr. William Farr, and Dr. Harry Crisp of NSWCDD; Dr. Carl Schmiedekamp of Naval Air Warfare Center; Evan Lock of Computer Command and Control Company; Nick Karangelen of Trident System Incorporated; Dr. Robert Goettge of Advance System Technology; Dr. Jane Liu and Dr. Kwei-Jay Lin of University of Illinois Urbana Champaign; Dr. Kane Kim of University of Carlifornia Irvine; Dr. Kishor Trivedi of Duke University; Dr. Geoffry Frank of Research Triangle Institute; Dr. Insup Lee of Penn State University; Dr. Osman Balci, Dr. James Author, and Dr. Richard Nance of Virginia Polytechnic Institute and State University; and everyone involved in this effort.

Approved by:

D.B. Colby

D. B. COLBY, Head
Systems Research & Technology Department

## ABSTRACT

The key to designing a real-time, large, complex system is to optimize the design to meet the requirements and desired measure of effectiveness. In order to achieve this, the system engineer/analyst must have the capability to specify the design goals/criteria, to quantify various aspects of the design, and to perform trade-offs among different design goals. One of the mechanisms that provides these capabilities is the System Design Factors. Whether the system design emphasis is on real-time, largeness, complexity, parallelism, or any specific criteria, it requires a set of System Design Factors to describe the properties, attributes and characteristics of the system. Each System Design Factor must have its own metric to gauge every detail of that system. The metric describes the weaknesses and strengths of a specific area in the design. In turn, the correlation of the System Design Factor characterizes the completeness and robustness of the system. Whether the system is designed top-down, bottom-up, or middle-out, the System Design Factors have major influence in design capture and analysis, design structuring decisions, allocation decisions, and trade-off decisions between various design structures and resource allocation candidates.

The main objectives of the System Design Factors research are to provide a) A mechanism to communicate from the customer to the development team throughout various phases of system engineering, b) A mechanism to quantify and identify a large, complex, real-time system's strengths and weaknesses so that effective comparison of different systems is achievable and c) A mechanism for linkage of various aspects of the design, which help the system engineer or analyst to specify, capture, analyze, design, prototype, test, evaluate, trade-off and implement the system effectively. This report presents a set of highly utilized System Design Factors that system engineers or analysts should consider early in the design to produce an effective system [HNH91], [HNH92].

# CONTENTS

## ILLUSTRATIONS

# CHAPTER 1

# INTRODUCTION

Traditionally, a system is built based on the needs of a customer. These needs are analyzed to determine the requirements and specifications [EdH91]. In turn, these requirements and specifications are captured to produce the initial design [Hoa91]. Analysis is executed to assure that the initial design is complete and consistent [BlF90], [Hoa91]. This design is optimized iteratively until a feasible or optimal design is achieved [HNH91], [HNH92]. Collected results are then passed through for rapid prototype, assessment, evaluation, test, and refinement to yield the final design [BoB85], [CYH91], [JeY91], [Kam91], [SvL76]. Implementation and test are then carried out to produce the final product, which is delivered to the customer. Many times, the customer will complain to the developer that the system did not meet the needs. The common causes for failing to meet the requirements might be one of the following: (a) the needs specified by the customer were not specific enough; (b) the needs were never clearly understood by the developers; or (c) communication among developers distorted the requirements as the development processes were performed.

The information understood by the whole system development team is crucial to produce a final product that meets the customer's needs. The current system engineering methodology lacks this communication mechanism from the customer to the whole development team.

The first objective of System Design Factors (SDF) research is to provide one such communication mechanism. In general, a system engineer or a customer wants some form to specify what criteria the end-result-system must meet. Depending on the desired criteria, it affects how the system would be designed and developed. These criteria are, in turn, the factors that the engineer must consider early in order to avoid bad designs, reduce cost, and optimize productivity [HHN90a], [HHN90b].

The second and third objectives are addressed by the following situation. Consider a situation where two system engineers were assigned to build a system independently given the same requirements and specifications from the same customer. When the two engineers delivered two systems to the customer, if the customer asks to compare quantitatively and qualitatively the different properties in term of performance, dependability, security, and real-time responsiveness of these two systems, then how does this comparison proceed. The second and third objectives of this research addressed this question. These objectives provide the mechanism for quantifying design goals of large, complex, real-time systems. With the current state of the system engineering technology, there are no normalized techniques to quantify and compare systems. If the system's properties could be specified quantitatively and qualitatively then its strengths and weaknesses can be identified and effective comparison among different systems can be achieved. Being able to qualitatively measure the system will not only benefit the system engineers for evaluation purposes, but it will also provide a benefit during the requirements specification phase, capture phase, analysis phase, design phase, optimization phase, and trade-off phase.

The focus of the SDF objectives are to provide a) A mechanism to communicate from the customer to the development team throughout various phases of system engineering, b) A mechanism to quantify and identify a large, complex, real-time system's strengths and weaknesses so that effective comparison of different systems is achievable and c) A mechanism for linkage of various aspects of the design, which help the system engineer or analyst to specify, capture, analyze, design, prototype, test, evaluate, trade-off, and implement the system effectively.

The proposed solution to these problems is to formulate hierarchical SDF. The short term goal is to collect concepts and ideas from government, industry, and academic sources to formulate a complete and robust system specification. The individual factors will be studied independently. The correlation of factors will be investigated. Testings and applications will be made to verify the correctness and consistency of the formulation. The long term goals are to refine the formulation, provide automation, and provide new system engineering mechanisms and concepts that will have significant impact on next generation of system engineering methodology.

The remainder of this document is organized as follows: Chapter 2: System Design Factors Taxonomy provides hierarchical view of SDF and provides current direction and focus of the research. Chapter 3: Example provides the touch and feel of SDF. Chapter 4: Specification and Use of SDF provides the utilization of the SDF template. Chapter 5: Current Status provides progress information; Chapter 6: Conclusion and Future Plans provide on-going research pursuit. Chapter 7: SDF Description provides the detail of each factor; and, finally, the Bibliography and Reference, Appendix, and SDF Working Group Members and points of contact.

# CHAPTER 2

# SDF TAXONOMY

The current thrust of this research is to define and formulate the SDF and their relationship. These factors are categorized. The formulation of these factors expresses the relationship and behavior of closely and loosely associated factors. The effect of the individual factors on the design or engineering process is being studied. The correlation of multiple factors is also undergoing study. The rating, normalizing, and voting techniques for these factors are being derived. The research is expected to generate a robust SDF taxonomy. Each factor will consist of terminology, definition, source, metrics, example, usage, and notes.

Currently there are 11 major groupings of factors that seem to be required for most large, complex, real-time systems. These groupings are arbitrary. Each of the groupings consists of factors that are closely associated with other factors, which ultimately affect the factor's behavior by inheritance. This hierarchical taxonomy will evolve as this research effort progresses. The current SDF taxonomy is shown below in Figure 2-1 to demonstrate the SDF framework. The detail description of this taxonomy is disclose in Chapter 7. This taxonomy provides a set of SDF that customers, system engineers, or an alysts should consider early in the design in order to produce an effective system [HNH91], [HNH92], [HHN90a], [HNH90b].

- 1 *PERFORMANCE*
  - 1.1 RESPONSE TIME
  - 1.2 CAPABILITY
  - 1.3 RELATIVE ACTIVITY
  - 1.4 SPEED
  - 1.5 THROUGHPUT
  - 1.6 LATENCY
  - 1.7 LOAD BALANCING
    - 1.7.1 INFORMATION LOAD
    - 1.7.2 PROCESSING LOAD
  - 1.8 GRACEFUL DEGRADABILITY / LOAD SHEDDING
  - 1.9 EFFICIENCY
  - 1.10 PREDICTABILITY
- 2 *REAL-TIME*
  - 2.1 HARDNESS
  - 2.2 HARD DEADLINES
    - 2.2.0.1 PERIODIC
    - 2.2.0.2 APERIODIC
    - 2.2.0.3 SPORADIC
  - 2.3 SOFT DEADLINES
    - 2.3.0.1 PERIODIC
    - 2.3.0.2 APERIODIC
    - 2.3.0.3 SPORADIC
  - 2.4 TEMPORAL DISTANCE
  - 2.5 TARDINESS
  - 2.6 NUMBER OF CONSECUTIVELY MISSED DEADLINES
  - 2.7 PREDICTABILITIES
  - 2.8 GRACEFUL DEGRADATION
- 3 *COMPUTATION/PROCESSING REQUIREMENTS*
  - 3.1 IMPORTANCE
  - 3.2 USEFULNESS
  - 3.3 PRIORITY
  - 3.4 (COMPUTING) PORTABILITY
  - 3.5 INTERRUPT/RESET CAPABILITIES
- 4 *DEPENDABILITY*
  - 4.1 RELIABILITY
  - 4.2 ACCURACY
  - 4.3 FAULT TOLERANCE
  - 4.4 GRACEFUL DEGRABILITY
  - 4.5 REDUNDANCY
    - 4.5.1 STATIC
    - 4.5.2 DYNAMIC
  - 4.6 AVAILABILITY
    - 4.6.1 INHERENT AVAILABILITY
    - 4.6.2 ACHIEVED AVAILABILITY
    - 4.6.3 OPERATIONAL AVAILABILITY
    - 4.6.4 EASE OF REPLACEMENT
    - 4.6.5 CRASH RECOVERABILITY
    - 4.6.6 COMPUTATION HEAVY PROCESS EFFECTS
  - 4.7 QUALITY
- 5 *SECURITY*
  - 5.1 CLASSIFICATION
  - 5.2 TYPE OF DATA
    - 5.2.1 LEVEL I (CLASSIFIED)
      - 5.2.1.1 TOP SECRET OR ABOVE
      - 5.2.1.2 SECRET
      - 5.2.1.3 CONFIDENTIAL
    - 5.2.2 LEVEL II (SENSITIVE)
      - 5.2.2.1 PRIVACY ACT/FINANCIAL
      - 5.2.2.2 FOR OFFICIAL USE ONLY
      - 5.2.2.3 SENSITIVE MANAGEMENT
      - 5.2.2.4 PROPRIETARY/PRIVILEGED
    - 5.2.3 LEVEL III (NONSENSITIVE)
      - 5.2.3.1 (OTHER--NOT CATEGORIES IN LEVEL I AND II)
  - 5.3 PERCENTAGE OF PROCESSING TIME / SECURITY LOAD
  - 5.4 ENCRYPTION TYPE REQUIREMENTS
  - 5.5 IMPLEMENTATION TECHNIQUES REQUIREMENTS
- 6 *HUMANWARE*
  - 6.1 EASE OF USE
  - 6.2 POTENTIAL OPERATOR DECISIONS
  - 6.3 1.OPERATOR DELAY / 2.USER RESPONSE TIME
  - 6.4 OPERATOR ACTION(S)
  - 6.5 1.REQUIRED NUMBER OF OPERATORS / 2.NUMBER OF SIMULTANEOUS USERS
  - 6.6 USER INTENSITY
  - 6.7 AVERAGE TIME FOR EACH CATEGORIES
  - 6.8 POTENTIAL ERRORS
- 7 *PHYSICAL REQUIREMENTS*
  - 7.1 SIZE REQUIREMENTS
    - 7.1.1 HEIGHT
    - 7.1.2 WIDTH
    - 7.1.3 LENGTH
    - 7.1.4 DEPTH
    - 7.1.5 AREA
    - 7.1.6 VOLUME
  - 7.2 WEIGHT REQUIREMENTS
  - 7.3 RUGGABILITY (RUGGEDIZED)
  - 7.4 SURVIVABILITY
  - 7.5 (PHYSICAL) PORTABILITY
  - 7.6 ENERGY REQUIREMENTS
    - 7.6.1 (ENERGY) CONSUMPTION
      - 7.6.1.1 ELECTRICAL (ENERGY CONSUMED)
      - 7.6.1.2 FUEL (ENERGY CONSUMED)
      - 7.6.1.3 OTHER (ENERGY CONSUMED)
    - 7.6.2 (ENERGY) DISSIPATED
  - 7.7 LOCATIONAL OPERATING ENVIRONMENT
    - 7.7.1 GEOGRAPHICAL LOCATION
    - 7.7.2 INDOORS/OUTDOORS
    - 7.7.3 TEMPERATURE
    - 7.7.4 HUMIDITY
    - 7.7.5 ACOUSTICAL NOISE
    - 7.7.6 AIR PURITY/QUALITY
    - 7.7.7 EXPOSURE TO WIND
    - 7.7.8 EXPOSURE TO WATER
    - 7.7.9 EXPOSURE TO ELECTROMAGNETIC RADIATION
    - 7.7.10 VIBRATIONS/STABILITY
  - 7.8 CLIMATE CONTROL
    - 7.8.1 COOLING
    - 7.8.2 HEATING
    - 7.8.3 HUMIDITY CONTROL
    - 7.8.4 ACOUSTICAL NOISE SUPPRESSION
    - 7.8.5 AIR PURITY/QUALITY CONTROL
    - 7.8.6 MOTION STABILIZATION
    - 7.8.7 LIGHTING
  - 7.9 MANUFACTURING CONSIDERATIONS
    - 7.9.1 PRODUCTION CAPACITY
    - 7.9.2 PRODUCTION TIME
  - 7.10 COMPUTER
    - 7.10.1 CPU
    - 7.10.2 MEMORY
    - 7.10.3 STORAGE
- 8 *FINANCIAL REQUIREMENTS*
  - 8.1 COST TO DEVELOP
  - 8.2 COST TO PROTOTYPE
  - 8.3 COST TO PRODUCE
  - 8.4 COST TO TEST
  - 8.5 COST TO PURCHASE
  - 8.6 COST TO OPERATE
  - 8.7 COST TO MAINTAIN
  - 8.8 COST TO REPAIR
  - 8.9 COST TO INCLUDE SECURITY CAPABILITY
  - 8.10 PRODUCTIVITY
- 9 *TIME PROJECTED*
  - 9.1 ESTIMATED TIME TO DEVELOP
  - 9.2 ESTIMATED TIME TO PROTOTYPE
  - 9.3 ESTIMATED TIME TO PRODUCE
  - 9.4 ESTIMATED TIME TO TEST
  - 9.5 ESTIMATED TIME TO PURCHASE
  - 9.6 ESTIMATED TIME TO OPERATE
  - 9.7 ESTIMATED TIME TO MAINTAIN
  - 9.8 ESTIMATED TIME TO REPAIR
  - 9.9 ESTIMATED TIME TO INCLUDE SECURITY CAPABILITY
- 10 *LIFE CYCLE*
  - 10.1 TESTABILITY
  - 10.2 MAINTENANCE
    - 10.2.1 EASE OF MAINTENANCE
    - 10.2.2 NUMBER OF PERSONS NEED TO MAINTAIN
    - 10.2.3 NOTIFICATION
    - 10.2.4 FREQUENCY
    - 10.2.5 MAINTENANCE DOWNTIME/DURATION
    - 10.2.6 DEGREE OF SYSTEM DISABILITY
    - 10.2.7 WHEN MAINTENANCE COMES DUE
    - 10.2.8 DURING MAINTENANCE
    - 10.2.9 WEAR LIFETIME
  - 10.3 OBSOLESCENCE LIFETIME
  - 10.4 REUSEABILITY
- 11 *FUTURE NEEDS CONSIDERATIONS*
  - 11.1 ADAPTABILITY/FLEXIBILITY
  - 11.2 EXPANDABILITY
  - 11.3 COMPATIBILITY
  - 11.4 INTERGRABILITY
  - 11.5 INTEROPERABILITY
  - 11.6 INTEGRITY

**FIGURE 2-1. TAXONOMY OF SYSTEM DESIGN FACTORS.**

# CHAPTER 3

## EXAMPLES

This section gives some small examples where SDF are used. Before this can begin, the characteristics structure must first be introduced. In order to effectively introduce the characteristics structure, some definitions are provided to give a common understanding.

**Quantitative Value** is a quantifiable measurement. It is a numerical value. It represents the degree of excellence. Some value may have a different type of range or minimum and maximum cardinality. For example, temperature could be measured as 120.5 degrees of Kelvin and could vary only between 0 and 277.15 degrees.
**Attribute** is the quality of a person or thing (non-physical).
**Property** is the attribute which belongs to some one or some thing (physical).
**Characteristics** is any special feature of a person or thing.

| Subject | Properties | Attributes | Quantitative Value | Qualitative Value (Characteristics) |
|---|---|---|---|---|
| | | | 10 to 50 | Slow |
| | | Air Speed | 51 to 75 | Moderate |
| | | | 76 to 100 | Fast |
| | | | 1 to 5 | Slow |
| | Performance | Land Speed | 6 to 10 | Moderate |
| | | | 11 to 15 | Fast |
| | | | 0.0 to 10.0 | Fast |
| | | Take-Off T. | 11.0 to 20.0 | Moderate |
| | | | 21.0 to 30.0 | Slow |
| | | | 0 to 1 | Good |
| | | Sickness T. | 1 to 3 | Average |
| | | | 6 to 10 | Poor |
| Eagle | Life Cycle | | | |
| | | | 0.0 to 5.0 | Short |
| | | Life Span | 5.1 to 10.0 | Medium |
| | | | 10.1 to 15.0 | Long |
| | | | 0.5 to 0.75 | Small |
| | | Size | 0.76 to 1.5 | Medium |
| | | | 1.6 to 2.0 | Large |
| | | | 1 to 5 | Brown |
| | Physical Req. | Color | 6 to 10 | Gray |
| | | | 11 to 15 | Black |
| | | | 1.0 to 10.0 | Short |
| | | Wing Span | 11.0 to 15.0 | Medium |
| | | | 16.0 to 20.0 | Long |

FIGURE 3-1. EXAMPLE OF CHARACTERISTICS STRUCTURE

The hierarchical relationship among these definitions forms a characteristics structure which provides a general mechanism for quantification. This mechanism is applied with the SDF to quantify systems. An example is given to demonstrate the relationship among these definitions. The example in Figure 3-1 shows hierarchically a **Subject** that has **Properties** which have **Attributes** which in turn have **Quantitative Value** and

**Qualitative Value (Characteristics).** Consider an eagle who has the following properties: performance, life-cycle, and physical requirements; performance which, in turn, has the following attributes: air speed, land speed, and take off time; life cycle which, in turn, has the following attributes: overall sickness time (health) and life span; physical requirements which, in turn, have the following attributes: size, color, and wing span; size which, in turn, has the following quantitative value (i.e., could vary between 0.5 to 2.0 feet) and qualitative value (characteristics) (i.e., could be small, medium, or large). The rest of the quantitative and qualitative values are shown in Figure 3-1.

In the above example, the subject was an eagle. However, the subject can be substituted with one of the following: system, subsystem, component, module, task, node, device, or any object. This characteristics structure provides a low level or detailed link to the criteria which, in turn, provides a high level link to the SDF. In other words, the characteristics structure applied to eagle allows us to quantify and rate different aspects of its species. A similar approach can be applied to the system, thereby allowing us to quantify and rate different factors of the system. The application of the characteristics structure to the SDF is demonstrated in Figure 3-2.

| Subject | Properties | Attributes | Quantitative Value | Qualitative Value (Characteristics) |
|---|---|---|---|---|
| or | or | or | | |
| Subject | Factors (Goal Oriented) | Associated Factors (Criteria/Decision Oriented) | Quantitative Value | Qualitative Value (Characteristics) |
| | | | 0.0 to 1.0 | Fast |
| | | Resp. Time | 1.1 to 2.0 | Medium |
| | | | 2.1 to 3.0 | Slow |
| | | | 1 to 5 | Slow |
| | Performance | Throughput | 6 to 10 | Medium |
| | | | 11 to 15 | Fast |
| | | | 2.1 to 3.0 | Slow |
| | | Latency | 1.1 to 2.0 | Medium |
| | | | 0.0 to 1.0 | Fast |
| | | | 0.5 to 0.8 | Bad |
| System, | | Reliability | 0.9 to 1.0 | Good |
| Subsystem, | Dependability | | | |
| Component | | | 0.0 to 1.0 | Bad |
| Object, | | Fault-Tol. | 1.0 to 2.0 | Moderate |
| etc. | | | 2.0 to 3.0 | Good |
| | | | 0.0 to 10.0 | Light |
| | | Weight | 11.1 to 20.0 | Medium |
| | | | 20.1 to 30.0 | Heavy |
| | | | 10 to 50 | Small |
| | Physical Req. | Size | 51 to 100 | Medium |
| | | | 101 to 150 | Large |
| | | | 0.0 to 1.2 | Low |
| | | Power | 1.3 to 2.5 | Moderate |
| | | | 2.6 to 3.8 | High |

FIGURE 3-2. EXAMPLE OF SYSTEM DESIGN FACTORS

As illustrated in this example (Figure 3-2), a customer may need to rate, measure, or design the system in terms of the following properties: performance, dependability [Joh85], [WaH91], and physical requirements.

Performance which, in turn, has the following attributes: response time, throughput, and latency. Dependability which, in turn, has the following attributes: reliability and fault-tolerance. Physical requirements which, in turn, have the following attributes: weight, size, and power. The response time could vary between 0.0 to 3.0 and its characteristics could be fast, medium, or slow. The rest of the quantitative and qualitative values are shown in the graph. This mechanism allows one to identify and effectively compare the strengths and weaknesses of different systems.

The next four short examples demonstrate the application of the factors. The first example is the application of a priority factor, the second is the weight factor, the third is the usefulness factor, and the fourth is the application of all three factors simultaneously. Weight and usefulness play an important role when it is being used in conjunction with priority. These three variables are used for the purpose of design structuring decisions, resource allocation decisions, scheduling decisions, and trade-off analysis in the general optimization method. The information provided in these examples may be in a proposal form, brainstorm state. They have not been proven or tested to be 100 percent correct. The information is meant for collaboration purposes. In each of the examples the factor is defined, the ranges are given, and the potential problem is pointed out.

## EXAMPLE 1

### Priority

Definition:

*Priority* emerges from the scheduling and operating system domain. It is commonly used as a ranking variable for determining when and where a task should be scheduled to meet its deadline.

Ranges:

Priority is defined as an integer value and its range is between 0 and the maximum number of tasks or modules within the system. Zero is defined as null or no priority while the maximum number of tasks is the highest priority.

Example:

A system is composed of 15 tasks and the priority is assigned as the following:

| Task Name | Calculated Priority Value | Priority Ranking |
|---|---|---|
| A | 0 | No Priority |
| B | Maximum-number-of-tasks -14 = 1 | Lowest |
| B1 | Maximum-number-of-tasks -13 = 2 | 2nd Lowest |
| ... | ... ... | |
| C | Maximum-number-of-tasks -2 = 1 | 3rd Highest |
| D | Maximum-number-of-tasks -1 = 1 | 2nd Highest |
| E | Maximum-number-of-tasks - 0 = 15 | Highest |

Problem:

A problem might occur in this type range when two systems are merging. This would cause a non-uniform priority scale or priority conflict. However, the idea of using the maximum number of tasks as the highest priority would allow the system size to expand and contract without having to reassign its priorities.

## EXAMPLE 2

### Weight

Definition:

*Weight* is defined as a variable assigned by the system designer, analyst, developer, or customer to emphasize the distribution of the individual criteria that he/she desired within the system.

Ranges:

Weight is defined as a real value and its range is between zero and one inclusively (i.e., [0.0,1.0]). Zero is defined as no emphasis and one, as heavily emphasized.

Example:

An overall system under design might be desired to emphasize the following criteria:

**Overall-System** = 0.2 * Performance + 0.3 * Dependability + 0.2 * Cost + 0.2 * Real-time + 0.1 * Security

where **Performance** = 0.3 * Communication + 0.3 * Computation + 0.2 * Response-time + 0.2 * Latency

where **Dependability** = 0.2 * Fault-tolerance + 0.8 * Reliability

where **Cost** = 0.5 * Maintain + 0.3 * Develop + 0.2 * Operate

where **Real-time** = etc...

where **Security** = etc...

This technique can be applied in a hierarchical fashion to subsystem, component, task, submodule, or devices, etc.; for instance, the tasks might be assigned with the following weights.

| Task's Name | Performance Weight | Dependability Weight | Security Weight | Real-Time Weight |
|---|---|---|---|---|
| A | 0.5 | 0.2 | 0.15 | 0.15 |
| B | 0.2 | 0.2 | 0.3 | 0.3 |
| C | 0.4 | 0.2 | 0.1 | 0.2 |
| D | 0.2 | 0.6 | 0.1 | 0.1 |
| E | 0.1 | 0.5 | 0.2 | 0.1 |

Problem:

A problem might occur in this type of range when two systems are merging. This would cause inconsistent weighing. However, the idea of using a normalized value would allow the system size to expand and contract without having to reassign its weight.

## EXAMPLE 3

### Usefulness

Definition:

*Usefulness* is defined as a variable assigned by the system designer, analyst, developer or customer to emphasize the criticality of individual component of the system.

Ranges:

Usefulness is defined as a real value and its range is between 0 and 100 inclusively (i.e., [1,100]). One defined as least useful and 100 as most useful.

Example:

A system was analyzed by the engineer, based on the functional criticality, and its usefulness is assigned as follows:

| Task's Name | Usefullness Value |
|---|---|
| A | 1 |
| B | 10 |
| C | 20 |
| D | 80 |
| E | 81 |

Problem:

A problem might occur in this range when two systems are merging. This would cause a non-uniform scale of usefulness. However, the idea of using usefulness would allow the system engineer to perform trade-off in the event of priority tie-breaking, weight tie-breaking, or both.

## EXAMPLE 4

### Applying Priority, Weight and Usefulness at Once

The individual c. .ınples given above seem to work fairly adequate. However, when the three variables are used together it is more complicated. They are used together to make trade-off decisions and for tie-breaking. One of the difficult tasks is to formalize a rule that will assist in making these types of decision. An example of a rule is: Usefulness value override and Weight value and Priority value, while the Weight value overrides the Priority value. One possible formulation for this is:

WUP-rating = Usefulness + Weight*Priority

Example: Given two tasks with the following usefulness, weight, and priority assignment.

| Task's Name | Usefulness Value | Performance Value | Priority Value |
|---|---|---|---|
| D | 80 | 0.2 | 14 |
| E | 50 | 0.1 | 15 |

Task D yields WUP-rating = 80 + 0.2*14 = 82.8
Task E yields WUP-rating = 81 + 0.1*15 = 81.5

Comparing WUP-rating of Tasks D and E

Although Task D has lower Usefulness and Priority values, its overall performance WUP-rating is higher, and therefore, the decision on performance should favor Task E. Other types (i.e., dependability, security, etc.) of WUP-ratings follow in a similar fashion.

The examples above demonstrate the application of various factors in different situations. The SDF allow the customer to specify the system hierarchy what factors are important to him, and acceptable or unacceptable results. This information helps the engineer to focus on specific criteria whether it is in the requirements phase, capture phase, design phase, analysis and evaluation phase, optimization and trade-off phase, or the implementation phase.

# CHAPTER 4

## SPECIFICATION AND USE OF SDF

The examples in Chapter 3 show the overall or top level application of the SDF. The detailed application of SDF is demonstrated through SDF Template (Figure 4-1). The purpose of this template is to provide a general format to guide the system engineer or the customer in the application of SDF. It assists the engineer/customer in specifying the goal/criteria to be measured and allows the template to be attached or probed onto a subsystem, a component, an object, or the whole system itself just as explained in the previous examples. This provides the metrification mechanism to quantify the various aspects of design.

| | | | |
|---|---|---|---|
| 1. | Name: | *Reliability of Beam Former* | |
| 2. | Type: | *Probability* | |
| 3. | Range: | *0.0 to 1.0* | |
| 4. | Units: | *Units of Probability* | |
| 5. | Methods/Principle: | *Fault Tolerance, Highly Reliable Component* | |
| 6. | Rationale: | *Life Critical Function* | |
| 7. | Relationship: | *Availability, Fault Tolerance, ...* | |
| | a. Relational Expression | *Positive Correlation, Negative Correlation* | |
| 8. | Quantification | | |
| | a. Type | | |
| | b. Formula | *R(t) = 1 - F(t)* | *Actual* |
| | | *.989 entered* | *Required* |
| | | *1.01 * Required* | *Budgeted* |
| 9. | Consistency Rule | | |
| | a. By aggregation | *Use Rule X and Y;* *Rule X: The probability of the component in series is the product of its probabilities.* *Rule Y: The probability of the component in parallel use one of rating, voting, scheme.* | |
| | b. By type | | |
| | c. By design factor | | |
| | d. By view | | |
| | e. By component | | |
| 10. | Reference: | *Author's name* | |
| 11. | Definition: | *Text Book* | |
| 12. | Annotation: | *Comments* | |
| 13. | Next Template: | | |

FIGURE 4-1. SDF TEMPLATE

The initial template was formulated and an example is given to get the touch and feel of the template. Currently, there are 13 items in this template. The Name item is a slot holder for the name of a specific design factor (e.g., # Reliability of Beam Former). The Type item is a slot holder for the classification of the factor (e.g., probability). The Range item is a slot holder for the minimum and maximum value or the cardinality of the factor (e.g., 0.0 to 1.0). The Units item is a slot holder for the unit of measurement of the factor (e.g., Units of Probability). The Methods/Principle item is a slot holder for the approaches or techniques that the designer/customer considered to be associated with this factor (e.g., Fault Tolerance, Highly Reliable Component). The Rationale item is a slot holder for the reason that this factor applies to a specific component/object (e.g., Life Critical Function). The Relationship item is the slot holder for the list of closely associated factors (e.g., Availability, Fault Tolerance). The Relational Expression field in this item provides the slot for the list of correlations between this factor and closely associated factors (e.g., Positive Correlation, Negative Correlation). The Quantification item contains the Type and Formula fields. The Type field in this item is the slot holder for either integer, float, double, short, or long. The Formula field in this item currently provides the slot for three mathematical expressions: (1) actually calculated (e.g., $R(t) = 1 - F(t)$ ), (2) required to be a specific value (e.g., 0.989), and (3) budgeted by designer or customer (e.g., 1.01 * 0.989). The Consistency Rule item consists of By-Aggregation, By-Type, By-Design Factors, By-View, and By-Component rules. For example, By-Aggregation field provides a slot that holds the rule for governing this factor consistently throughout the hierarchy (e.g., Use Rule X and Rule Y). The Reference item is a slot holder for the source of reference or the name of the author that formulated this factor. The Definition item is a slot holder for the clarity for this factor. The Annotation Item is the slot holder for commenting on relevant information or providing warnings related to this factor. Lastly, the Next Template item is not completely defined at this time, but it is the slot holder for any detailed specification that may not require the customer's direction.

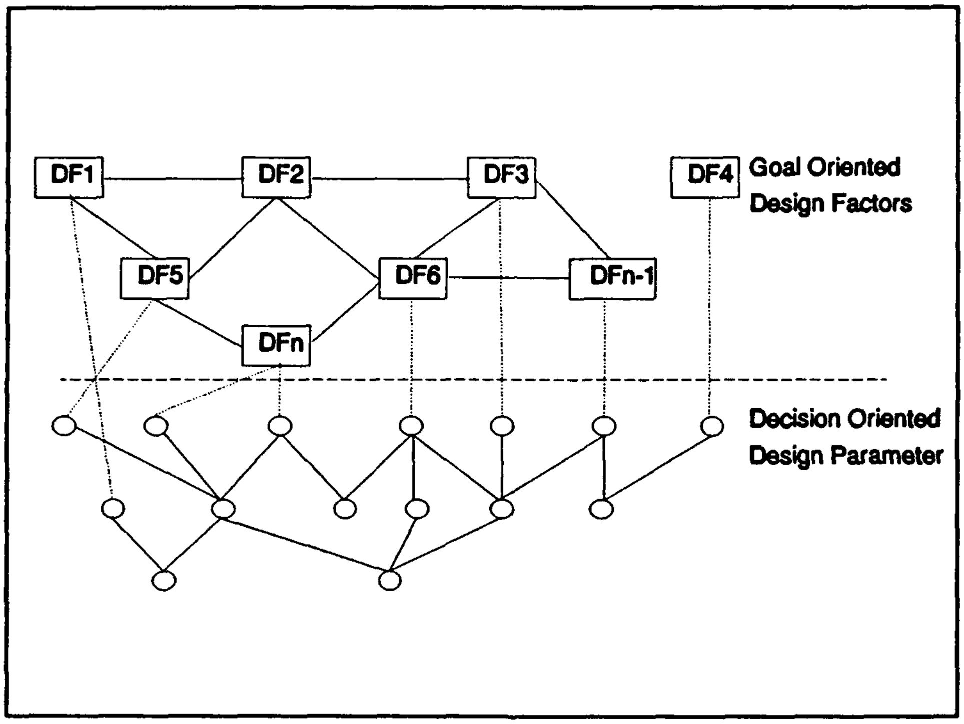


FIGURE 4-2. DESIGN FACTORS DEPENDENCIES GRAPH

The advantage of this template is not just to ease the use of the factors but it also allows the designer/customer to take the available factors and customize his own design factors appropriate for his specific needs. It is up to the engineer/customer to decide the important and unimportant factors and formulate the design goal and design decision that the end-result system must meet. The overall design goal and design decision of the system can be described by the SDF dependencies graph shown in Figure 4-2.

The upper half of the graph is referred to as the goal oriented design factors, while the lower half is referred to as the decision oriented design parameters. The goal oriented is independent of the implementation model while the decision oriented is dependent on the implementation model. It would be ideal for the design to be implementation independent in the design phase; however, in practice it is not always the case. SDF dependencies provide the linkage between the implementation independent (Design Goal) and implementation dependent (Design Parameter). The SDF dependencies graph assists the engineer/customer in understanding the behavior change of the individual factor. These changes are based on its' closely and loosely associated factors. The behavior of each subsystem, component, object, or the whole system with respect to different factors (design goals) can be analyzed separately or simultaneously.

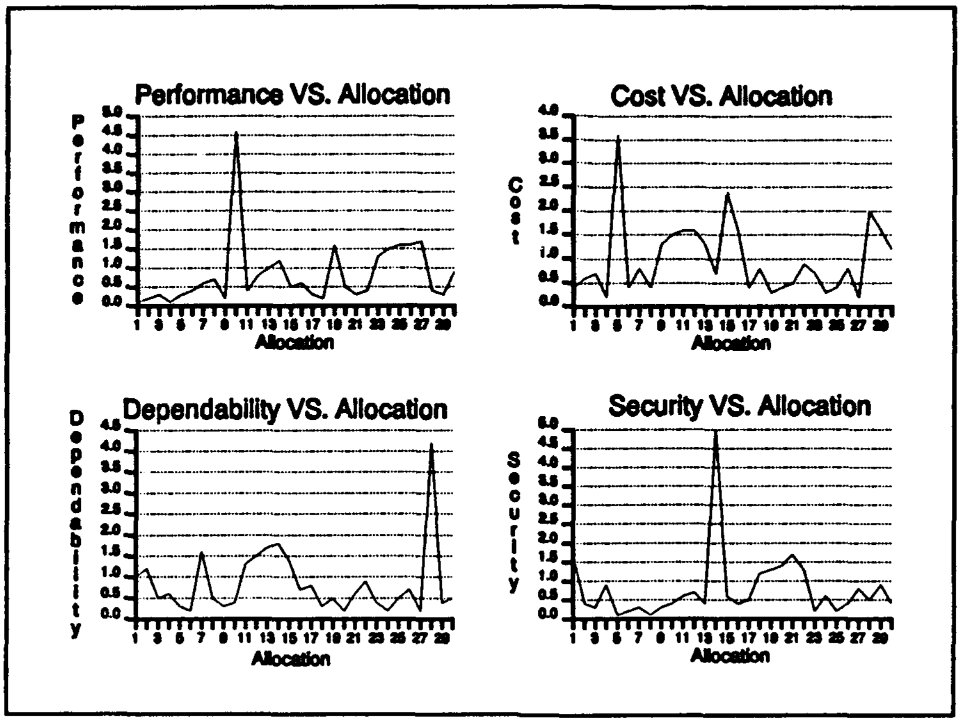


FIGURE 4-3. SINGLE CRITERIA OBJECTIVE FUNCTION

Although the scope of this paper is not to cover *Design* Structuring and Allocation Optimization methodology, it is worth showing some applications of SDF with such a method [HHN90a], [HHN90b], [HNH91], [HNH92]. Assume that a customer procured a contractor to develop a system such that the end-result system is required to meet certain measurements in terms of Performance, Dependability, Cost, and Security. As illustrated with the previous template example, the engineer can specify and attach these required factors to the design. Based on the design goal and design parameter, the engineer can tailor the single criteria or multi-criteria objective function for optimization [NaF91].

This design is then optimized based on the tailored objective function. The first approach that the engineer could take is to optimize the design with a single criteria objective function (shown in Figure 4-3) and then overlay the result (shown in Figure 4-4) to execute trade-off analysis [Dos91]. The second approach would be to optimize the design with multi-criteria objective function (shown in Figure 4-5). The first approach optimizes the criteria one at a time, while the later approach optimizes these criteria simultaneously.

The results of single and multi-criteria objective function together with the SDF dependencies graph provide the engineer with a better understanding of the system under design. By understanding the physical nature or correlation among the factors, the designer/customer can predict the behavior and perform effective trade-off. The application of the SDF with optimization here merely demonstrates some utility of the SDF. SDF can be applied throughout various phases of system engineering. It is a critical component in system engineering.

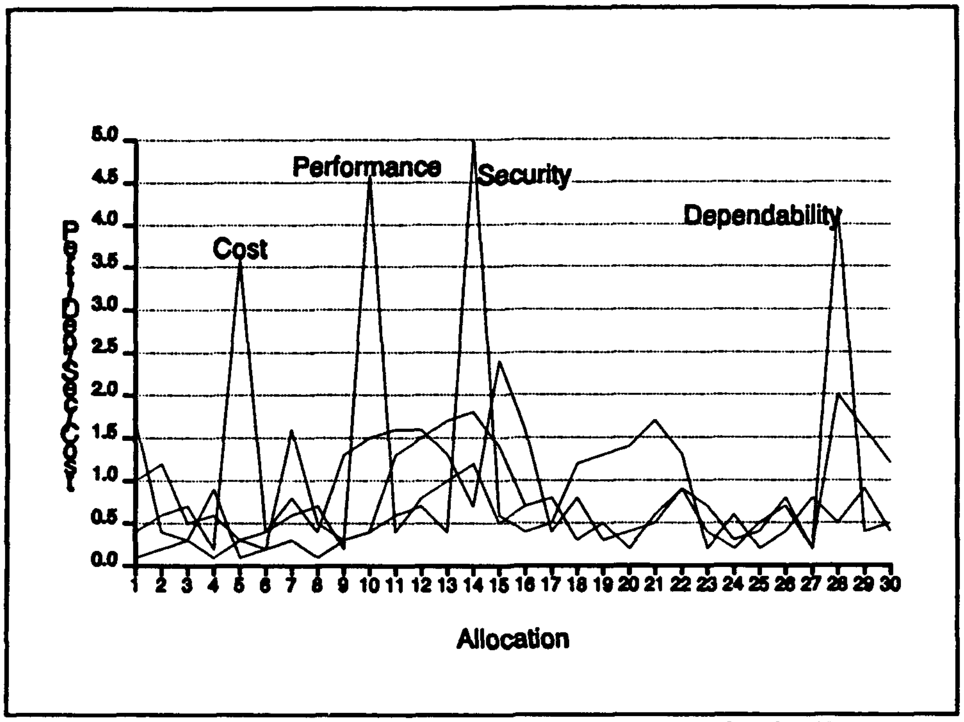


FIGURE 4-4. SINGLE CRITERIA OBJECTIVE FUNCTION OVERLAY

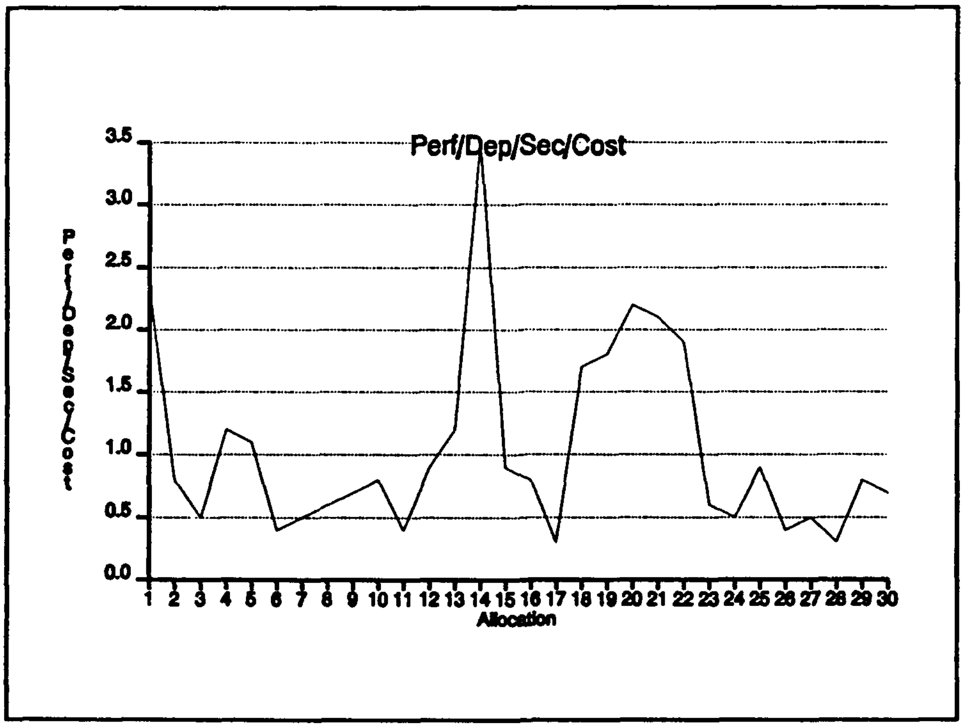


FIGURE 4-5. MULTI-CRITERIA OBJECTIVE FUNCTION

# CHAPTER 5

## CURRENT STATUS

A list of SDF was generated and structured in the taxonomy format. There are 11 main groupings of factors and their closely associated factors defined so far. Currently, the relationship of these factors is not well understood but we are attempting to correlate these factors as this effort progresses. Although part of this document is not fully completed in this version, this draft provides a detailed description of many design factors. The description consists of the terminology, definition, source, metrics, example, usage, and note. The terminology provides the commonly used vocabulary word. The definition provides the meanings of the factor. The source provides the reference of the definitions. The metrics [JuA91] provide the unit of measurement (dimension) of the factor. The example provides an illustration of the factor. The usage provides cases when, where, how, and why to apply the factor. Lastly, the note provides any relevant information or warning related to the factor. An initial SDF template and example were demonstrated to get the feel of the formulation. The prototyping of the SDF template is underway. An initial SDF focus group has been established to collaborate and to clarify issues in the SDF formulation.

# CHAPTER 6

## CONCLUSION AND FUTURE PLANS

The goal of this effort is to generate a list of SDF. These factors are intended to be used throughout the entire system engineering process. For instance, they are used to specify in the requirements phase, encapsulate in the capturing phase, quantify and evaluate in the analysis phase, characterize in the optimization phase, and justify in the design trade-off phase. These factors are critical to the system engineering process.

Future plans include refining, restructuring, and streamlining (if necessary) the SDF. More dedicated research is being considered to focus on a smaller but widely used set of design factors. From this smaller set of design factors, intensive correlation will be studied. The formulation will be incorporated into the sonar example [Hoa91] and the Destination Level I Prototype [HNH92], [HNH91] in other research efforts for testing and refining.

The lessons learned in this effort will benefit the systems engineering community. The list is expected to evolve as the effort progresses. This is a collaborative effort among Naval Surface Warfare Center (NSWCDDWODET), DoD, other Government agencies, industry, and university communities.

# CHAPTER 7

## SDF DESCRIPTION

This section provides the description of each design factor. The description consists of the terminology, definition, source, metrics, example, usage, and note. The terminology provides the common use vocabulary word. The definition provides the meaning of the factor. The source provides the reference to the factor. The metrics provide the unit of measurement (dimension) of the factor. The example provides an illustration of the factor. The usage provides the cases of when, where, how, and why to apply the factor. Lastly, the note provides any relevant information or warning related to the factor. The description is described in the following grouping order:

1. PERFORMANCE
2. REAL-TIME
3. COMPUTATION/PROCESSING REQUIREMENTS
4. DEPENDABILITY
5. SECURITY
6. HUMANWARE
7. PHYSICAL REQUIREMENTS
8. FINANCIAL REQUIREMENTS
9. TIME PROJECTED
10. LIFE CYCLE
11. FUTURE NEEDS CONSIDERATIONS

# PERFORMANCE FACTOR

**TERM: Performance**

DEF:

1. Performance is usually specified or measured as either the time it takes to perform a critical function or as an execution rate of a basic operation.
2. Performance is a measurement which consists of a weighted sum of various variables of interest such as computation speed, computation power, communication speed etc.
3. Performance is an element of the specification which can be quantified in terms of either throughput or response time.
4. Performance is a measure or group of measures that quantifies the ability of the system to do what is required of it.
5. Performance is the effectiveness with which the resources of the host system are utilized toward meeting the objectives of the system. This definition can be paraphrased as, "How well does the system do what it is intended to do?"

SOURCE:

1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 321).
2. NSWCDD, CMN.
3. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 82).
4. NSWCDD, CJW(dict)
5. Svobodova, Liba, Computer Performance Measurement and Evaluation Methods: Analysis and Applications, 1976, p.8.

METRICS:

1. Time unit such as seconds, micro seconds, etc.

EXAMPLE:

1. Assume that data are transferred and transformed in a total time $T^d$ and a total time $T^c$ is required for control. These two times are further subdivided as follows:

Data Flow, $T^d$:

- Data transfers: $T^d_{tr}$
- Data Transformation: $T^d_{tm}$

Control, $T^c$:

- Data transfer control: $T^c_{tr}$
- Data transformation control: $T^c_{tm}$

In the worst case, each of these operations is done sequentially so that the minimum time for an operation involving a data transformation (e.g., multiply two numbers) is

$$T_{min}(\text{sequential}) = T^d_{tr} + T^d_{tm} + T^c_{tr} + T^c_{tm} \cdot$$

If, in the other extreme case, each of these operations could be executed in parallel, then

$$T_{min}(\text{parallel}) = \max[T^d_{tr} + T^d_{tm} + T^c_{tr} + T^c_{tm}]$$

In either case, the performance of a particular processor (or a subcomponent) can be computed as either

$$\frac{operation}{second} = \frac{1}{T_{min}}$$

or response time

$$TR = T_{min} \text{ x total number of operations}$$

If an algorithm consists of a sequence of operations which execute with a different

$$T_{min_i},$$

the model is easily extended to

$$TR = \sum_{1}^{n} T_{min_i} \times \text{number of operations of type } i$$

2. A computer system can execute certain instructions per seconds and can provide the amount of effective result within a certain ume.

USAGE: 1. N/A

2. This measurement is used to rate or compare the effectiveness (either fast or powerful or both) of a product from different manufacturers or vendors.

NOTE: 1. Refer to response time, speed, throughput, latency, communication, computation

**TERM: Response Time**

DEF: 1. The time necessary to carry out a task, job, or assignment (i.e. from the time it initiates to the time it completed).

2. The response time is equal to the time to execute an operation of type i multiplied by the number of operations of type i.

$$TR = \sum_{1}^{n} T_{min_i} \times \textit{number of operations of type i}$$

3. The amount of time it takes to react or reply to a request made upon the system. This time usually, but not necessarily, includes the amount of time to complete the request or task.

4. Elapsed time between submitting requests and transactions and receiving their output in an interactive or real time system.

SOURCE: 1. NSWCDD, CMC.

2. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 321).

3. NSWCDD, CJW.

4. Svobodova, Liba, Computer Performance Measurement and Evaluation Methods: Analysis and Applications, 1976, p.16.

METRICS: 1. Times unit such as minutes, seconds, microseconds, etc.

EXAMPLE: 1. Refer to performance definition and example

3. The ambiguity in the definition can be seen in the example of requesting a computer printout. Response time could be defined as the point when the computer acknowledges that printing has begun, or when the computer finishes dumping the request into a print buffer, or when the actual hardcopy printout is complete; etc.

USAGE: 1.

NOTE: 1. Refer to performance definition and example. Refer to turn around time

**TERM: Relative Activity**

DEF: 1. Relative activity, $r_k$, is the ratio of the total time of an activity, $a_k$, and the total elapsed time.

2. Relitive activity is frequently used as a performance measure (CPU utilization, channel utiiization). It measures the time spent performing a particular activity during a particular time.

SOURCE: 1. Svobodova, Liba, Computer Performance Measurement and Evaluation Methods: Analysis and Applications, 1976, p.78-79.

METRICS:

$$\textbf{Relative Activity} = r_k = \frac{1}{t-t_0}\int_{t_0}^{t} a_k(\tau)\, d\tau$$

*where $t_0$ and $t$ are the starting and finishing times of event, $a_k$, respectively,*
*and*
*$a_k$=1 if it's a possible event for the system state, and is 0 otherwise.*

EXAMPLE: 1. N/A

USAGE: 1. N/A

NOTE: 1. N/A

**TERM: Capability**

DEF: 1. A measure of the computing capacity limits of the system.

SOURCE: 1. Svobodova, Liba, Computer Performance Measurement and Evaluation Methods: Analysis and Applications, 1976, p.16.

METRICS: 1. (Maximum amount of useful work that can be performed with a given workload) ÷ (unit of time)

EXAMPLE: 1. System X is capable of compiling 100 lines of source code per minute.

USAGE: 1. N/A

NOTE: 1. N/A

**TERM: Speed**

DEF: 1. A measure of how quickly the system runs or operates in a general sense.

SOURCE: 1. NSWCDD, CJW

METRICS: 1. Will be system dependent, but some examples of typical computer system speed metrics are: MIPS (Millions of Instructions Processed a Second), FLOPS (number of FLOating Point calculations a Second), and processor clock frequency.

EXAMPLE: 1. N/A

USAGE: 1. N/A

NOTE: 1. N/A

**TERM: Throughput**

DEF: 1. Throughput can be specified in two categories: first, Computational: the number of calculations or the number of the processes executed per unit time; and/or; second Communications: The number of information elements being communicated per unit time.
2. A measure of computation speed. Throughput is a measure of how quickly the results of a particular process can be periodically obtained.
3. That which enters the system in one form and leaves the system in another.

SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 82).
2. NSWCDD, CJW
3. Blanchard and Fabrycky, Systems Engineering and Analysis, 1990, p.4.
4. Svobodova, Liba, Computer Performance Measurement and Evaluation Methods: Analysis and Applications, 1976, p.16.

METRICS: 1. N/A
2. Minutes, seconds, microseconds, etc.
3. (Amount of useful work completed with a given workload) ÷ (unit of time)

EXAMPLE: 1. N/A

*Consider a processor that can execute C instructions per second . If a processor $P_i$ requires $X(i)$ instructions , then*

*1. The response time for each process is*

$$t_i = \frac{X(i)}{C}$$

*2. If N processes are executed, the total time is*

$$T = \sum_{i=1}^{N} \frac{X(i)}{C}$$

*and the average process throughput is $\frac{N}{T}$ , disregrarding overhead.*

2. See example #1 for the term LATENCY.

USAGE: 1. N/A

NOTE: 2. See note #1 for the term LATENCY.

**TERM: System Throughput**

DEF: 1. The ideal growth in system throughput, as more processors are added, is a straight line function, i.e.,

$$System\ Throughput\ (MIPS) = number\ of\ processor \times \frac{MIPS}{processor}$$

SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 83).
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Latency**
DEF: 1. A measure of computation speed. Latency is the amount of time from when a particular process or computation is initiated to when its final results are made available.
SOURCE: 1. NSWCDD, CJW
METRICS: 1. Minutes, seconds, microseconds, etc.
EXAMPLE: 1. Suppose a computation must pass serially through three processes, each with a processing time of 10sec, to determine a result. Now assume that these three processes are ideally pipelined. The throughput of such a computation would be 10sec while the latency would be 30sec.
USAGE: 1. N/A
NOTE: 1. For nonparallel, non-pipelined systems, latency and throughput will be equal; for parallel, pipelined systems, latency will generally be greater than throughput.

**TERM: Efficiency**
DEF: 1. A measure of the effectiveness of the system. Efficiency is a rating of the amount of "output" from a system as compared to the amount of "input" or maximum possible output.
2. In general, efficiency is the total quantifying output divided by the total quantifying input.
SOURCE: 1. NSWCDD, CJW(dict), and Svobodova, Liba, Computer Performance Measurement and Evaluation methods: Analysis and Applications, 1976, p.9.
2. NSWCDD, CMN
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Computational Loading**
DEF: 1. To estimate the computation loading by the over all system. It is essential to work from the bottom up. Beginning with the lowest level of decomposition of the data flow graphs, calculate the computation load for each data value transformation node. This is done by calculating or estimating the number of the arithmetic operations associated with one iteration of the individual node and dividing by the time budget to get an estimate of node computational load in operations per seconds.
SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 131).
METRICS: 1. operation per seconds
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Predictability**
DEF: 1. The extent to which a system behaves as expected by the user, designer, another system, etc.
SOURCE: 1. NSWCDD, CJW
METRICS: 1. N/A

| | |
|---|---|
| EXAMPLE: | 1. N/A |
| USAGE: | 1. N/A |
| NOTE: | 1. N/A |

# REAL-TIME FACTOR

**TERM: Real-Time**
DEF: 1. N/A
SOURCE: 1. N/A
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Deadlines**
DEF: 1. The time at or before a particular computation, response, task, etc. must be completed
SOURCE: 1. UIUC, JWL
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Hard Deadlines**
DEF: 1. A deadline where what is required to be done is absolutely completed in the specified time.
2. A deadline that has catastrophic effects on the system if it is not met.
SOURCE: 1. NSWCDD, CJW
2. NSWCDD, CMN
METRICS: 1. minutes, seconds, microseconds, etc.
EXAMPLE: 1. An anti-torpedo response system must complete its task in no less than X sec or else the torpedo sinks the boat.
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Soft Deadlines**
DEF: 1. A deadline where it is acceptable to take longer than the specified time as long as the deadline is met "on average" or on some similar sort of criteria.
2. A deadline that has a debilitating or degrading effect on the system if it is not met.
SOURCE: 1. NSWCDD, CJW
2. NSWCDD, CMN
METRICS: 1. Time of deadline in minutes, seconds, micro-seconds, etc. AND Condition on how deadline is to be met.
EXAMPLE: 1. On the average, an air-conditioner controller needs an update on room temperature every minute to maintain the temperature within X degrees of the preset.
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Temporal Distance**
DEF: 1. This is the duration within which two related tasks must be completed. Temporal distance can be maximum distance or minimum separation.
SOURCE: 1. UIUC, JWL
METRICS: 1.
EXAMPLE: 1. An example of maximum temporal distance in a passive sonar system is when the display of a target and the activation of the audio is required to be within 100 msec. An example of the separation is in traffic control. Aircrafts take off no closer than 2 minutes apart....
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Tardiness**
DEF: 1. The tardiness of a task when the deadline has been completed is the time from the deadline of the task to the completion time of the task. We sometimes want to require the tardiness is less than a certain amount. It is often acceptable, even for tasks with hard deadlines, if

deadlines are missed only by a little. Also, for some tasks with soft deadlines, late results are acceptable only when their tardiness are small.

SOURCE: 1. UIUC, JWL
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Number of consecutively missed deadlines**

DEF: 1. For the periodic task, this measures the number of requests in a row that are not completed on time. An occassional missed deadline is often acceptable, again even for periodic tasks with hard deadlines. e.g., control law computations. The task is considered a fatal failure, e.g. the system becomes unstable, when 5 or 6 consecutive deadlines are missed.

SOURCE: 1. UIUC, JWL
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

# COMPUTATION/PROCESSING REQUIREMENTS FACTOR

**TERM: Computation/Processing Requirements**
DEF: 1. Characteristics and conditions upon the calculations a system is to compute and the information and data the system is to process.
SOURCE: 1. NSWCDD, CJW
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Importance**
DEF: 1. A measure of criticality or significance. Importance relates heavily to both usefulness and priority. The importance is the necessity of the process, system, computation, etc., that is being described.
SOURCE: 1. NSWCDD, CJW
METRICS: 1. N/A
EXAMPLE: 1. A missile detection system would conceivably be a very "important" system even though it would probably be in use for a limited amount of time.
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Usefulness**
DEF: 1. A measure of utility and practicality. Usefulness relates heavily to both importance and priority. Usefulness tells of the value of the process, system, computation, etc., that is being described.
SOURCE: 1. NSWCDD, CJW
METRICS: 1. N/A
EXAMPLE: 1. A living quarters air conditioning system would probably be a very "useful" system even though it would probably not hold a great deal of "importance" on a warship, for example.
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Priority**
DEF: 1. A ranked description of the precedence to be given under particular conditions to the process, system, computation, etc., that is being described.
2. Priority assigned to a job by the user.
SOURCE: 1. NSWCDD, CJW
2. Svobodova, Liba, Computer Performance Measurement and Evaluation Methods: Analysis and Applications, 1976, p.12.
METRICS: 1. Classifying scheme that indicates a rank among what is being described.
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: (Computing) Portability**
DEF: 1. The ease with which things such as operating systems, system platforms, and application software can be changed or used on different systems, if at all.
SOURCE: 1. NSWCDD, CJW
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Interrupt/Reset Capabilities**
DEF: 1. A description of the abilities to "break in" or reset system processes at various points in the system's operation.

SOURCE: 1. NSWCDD, CJW
METRICS: 1. N/A
EXAMPLE: 1. This will be system dependent; some examples are turning the system off being the only way to reset the system, to a set of interrupt controls or commands applicable to particular interrupt situations.
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Memory Space**
DEF: 1. The type, format, and quantity of storage capacity needed for the system.
SOURCE: 1. NSWCDD, CJW
METRICS: 1. For a computer system, memory space is typically measured in bytes of Random Access (RAM) and Read Only Memory (ROM).
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

# DEPENDABILITY FACTOR

**TERM: Dependability**

DEF: 1. The quality of the service delivered such that the service is justifiably reliable.

SOURCE: 1. J. Laprie, "Dependable Computing and Fault Tolerance Concepts and Terminology," FTCS, 1985.

METRICS: 1. N/A

EXAMPLE: 1. N/A

USAGE: 1. N/A

NOTE: 1. N/A

**TERM: Reliability**

DEF: 1. The probability of performing the operational role for a specified time. It is a function of many factors including the model assumed for failure mechanisms. It is characterized by a mean-time-between-failures (MTBF) prediction.

2. The quality describing the degree to which the system exhibits a lack of probable failure and/or error.

3. The probability that a system or product will perform in a satisfactory manner for a given period of time when used under specified operating conditions.

4. The reliability R(t) of a system is a function of time, defined as the conditional probability that the system will perform correctly throughout the interval $[t_0,t]$, given that the system was performing correctly at time $t_0$. In other words, the reliability is the probability that the system will operate correctly throughout a complete interval of time. The reliability is a conditional probability in that it depends on the system being operational at the beginning of the chosen time interval. Reliability is most often used to characterize systems in which even momentary periods of incorrect performance are unacceptable, or in which it is impossible to repair the system. If repair is impossible, such as in many space applications, the time intervals being considered can be extremely long, perhaps as many as 10 years. In other applications, such as aircraft flight control, the time intervals of concern can be no more than several hours, but the probability of working correctly throughout that interval can be 0.9999999 or higher. It is a common convention when reporting reliability numbers to use $0.9_i$ to represent the fraction that has i nines to the right of the decimal point. For example, 0.9999999 is written as $0.9_7$.

SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 84).

2. NSWCDD, CJW and W. Beam, Systems Engineering - Architecture and Design, 1990, p.136.

3. Blanchard and Fabrycky, Systems Engineering and Analysis, 1990, p.347 and p.351 (The Reliability Function).

4. Barry. W. Johnson, Design and Analysis of Fault Tolerant Digital Systems, p. 4, Addison-Wesley Publishing Company, 1985.

METRICS: MTTF - Mean Time to Failure.

MTBF - Mean Time Between Failures (Beam p.136)

$\lambda$ - failure rate

note: MTBF = $^1/_\lambda$

Both MTBF and MTTF are measured in an appropriate time unit such as days, hours, minutes, etc.

*The Reliability Function*

The reliability function, also know as the survival function, is determined from the probability that a system will be operational at least for some specified time *t*. The reliability function, *R(t)*, is defined as:

$$R(t) = 1 - F(t)$$

where *F(t)* is the probability that the system will fail by time *t*. *F(t)* is basically the failure distribution function or unreliability function. If the random variable *t* has a probability density function *f(t)*, the expression for the reliability function is:

$$R(t) = 1 - F(t) = \int_t^{\infty} f(t)\,dt$$

If the time to failure is distributed according to an exponential density function, then:

$$f(t) = \frac{1}{\theta}e^{-\frac{t}{\theta}} \quad \rightarrow \quad R(t) = \int_t^{\infty} \frac{1}{\theta}e^{-\frac{t}{\theta}}\,dt = e^{-\frac{t}{\theta}}$$

where $\theta$, Mean life, is the average of the lifetimes of all items considered, which for the exponential distribution is MTBF. Thus:

$$R(t) = e^{-\frac{t}{M}} = e^{-\lambda t} \qquad \theta = M = \frac{1}{\lambda}$$

$\theta$ = *Mean life*
$M$ = *MTBF - Mean Time Between Failure*
$\lambda$ = *failure rate*

EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**Term: Unreliability**
DEF: 1. The **unreliability** Q(t) of a system is a function of time, defined as the conditional probability that a system will perform *incorrectly* during the interval $[t_0,t]$, given that the system was performing *correctly* at time $t_0$. The unreliability is often referred to as the *probability of failure.*
SOURCE: 1. Barry W. Johnson, Design and Analysis of Fault Tolerant Digital Systems, p. 4, Addison-Wesley Publishing Company, 1985.
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Accuracy**
DEF: 1. The degree to which a measurement conforms to its true or standard value.
SOURCE: 1. NSWCDD, CJW(dict)
METRICS: 1. There are many accepted standards of accuracy measurement. For example, specifying a figure to be within plus or minus a certain absolute amount or to be within a certain percentage of a stated quantity.
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Fault Tolerance**
DEF: 1. Fault tolerance is the ability of the system to continue operations in the presence of a failure (either software or hardware) without human intervention; ideally, the MTTR is zero. In distributed systems, fault tolerance usually implies the operations will continue in the same topological configuration with the same performance while the fault is discovered and repaired. Mechanisms to accomplish this include redundancy plus fault detection and reconfiguration hardware.
2. The capability where one or more functional parts of the system can fail without causing complete system failure.
SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 85).
2. Beam, W., Systems Engineering, p.140.
METRICS: 1. N/A

EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. Refer to redundancy, static redundancy, dynamic redundancy, and P11 task 5 dictionary.

**TERM: Graceful Degradation**
DEF: 1. Graceful degradation is used to describe the ability to continue operation in some degraded, but acceptable, mode in the presence of fault. This attribute must often be demonstrated as a part of the system acceptance.
SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 85).
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Redundancy**
DEF: 1. Redundancy can be incorporated in two ways: static and dynamic. In both cases, mechanisms to recover from the effect of the failure(e.g., loss of data or unacknowledged data) must be in place.
SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 85).
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. Refer to Static Redundancy, Dynamic Redundancy and P11 Task 5 dictionary.

**TERM: Static Redundancy**
DEF: 1. Static redundancy can be achieved by the duplication of components both of which operate simultaneously. Disagreements must be detected and the faulty unit identified (either by self-, mutual, or external sanity checks) and removed. The operation unit continues to operate while the faulty one is repaired. MTTR can be minimized by efficient error detection and fault identification routines plus a spares policy that allows quick replacement. Triple redundancy can be used to shorten the search time for a faulty unit. To accomplish this, three units operate in a majority vote environment. A disagreeing unit is immediately removed, while the other two continue to function. Static redundancy is conceptually simple to impose on a system; however, it is expensive if widely applied, since it involves duplicating components plus the necessary additional interface hardware and software.
SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 85).
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. Refer to Redundancy, Dynamics Redundancy and P11 Task 5 dictionary.

**TERM: Dynamic Redundancy**
DEF: 1. Dynamics redundancy implies the selective replacement of failed components or the reconfiguration of the system so that operation can continue. This implies mechanisms to detect the failure and either to integrate the appropriate spare component or reconfigure the system into a degraded but still useful operational mode.
SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 85).
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. Refer to redundancy, static redundancy and P11 Task 5 dictionary.

**TERM: Availability**

DEF: 1. The percentage of time the system is operational.

$$A = \frac{MTBF}{MTBF + MTTR} \times 100\%$$

where MTBF is Mean Time Between Failure
and MTTR is Mean Time To Repair

2. The portion of the time during which the system is able to be operated, of the entire time during which it is required to be operable.

3. Availability A(t) is a function of time, defined as the probability that a system is operating correctly and is available to perform its functions at the instant of time t. Availability differs from reliability in that reliability depends on an *interval* of time, whereas availability is taken at an *instant* of time. A system can be highly available yet experience frequent periods of inoperability as long as the length of each period is extremely short. In other words, the availability of a system depends not only on how frequently it becomes inoperable but also on how quickly it can be repaired. The most common measure of availability is the expected fraction of time that a system is available to correctly perform its functions.

SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 84).

2. W. Beam, Systems Engineering.

3. Barry W. Johnson, Design and Analysis of Fault Tolerant Digital Systems, Addison-Wesley Publishing Company, 1985, p. 5.

METRICS: 1. MTTR-Mean Time To Repair (Beam p.136)

EXAMPLE: 1. N/A

USAGE: 1. N/A

NOTE: 1. N/A

**TERM: Inherent Availability ($A_i$)**

DEF: 1. The probability that a system or equipment, when used under stated conditions in an *ideal* support environment (i.e., readily available tools, spares, maintenance personnel, etc.), will operate satisfactorily at any point in time as required.

SOURCE: 1. Blanchard and Fabrycky, Systems Engineering and Analysis, 1990, p.359.

METRICS:

$$A_i = \frac{MTBF}{MTBF + MTTR}$$

MTBF = Mean Time Between Failure
MTTR = Mean Time to Repair

EXAMPLE: 1. N/A

USAGE: 1. N/A

NOTE: 1. N/A

**TERM: Achieved Availability ($A_a$)**

DEF: 1. The probability that a system or equipment, when used under stated conditions in an *ideal* support environment will operate satisfactorily at any point in time. This definition differs from that of INHERENT AVAILABILITY in that preventive (i.e., scheduled) maintenance is included.

SOURCE: 1. Blanchard and Fabrycky, Systems Engineering and Analysis, 1990, p.359.

METRICS:

$$A_a = \frac{MTBM}{MTBM + \overline{M}}$$

MTBM = Mean Time Between Maintenance
$\overline{M}$ = Mean active maintenance time

USAGE: 1. N/A
EXAMPLE: 1. N/A
NOTE: 1. N/A

**TERM: Operational Availability ($A_o$)**
DEF: 1. The probability that a system or equipment, when used under stated conditions in an *actual* operational environment, will operate satisfactorily when called upon.
SOURCE: 1. Blanchard and Fabrycky, Systems Engineering and Analysis, 1990, p.359.
METRICS:

$$A_0 = \frac{MTBM}{MTBM + MDT}$$

*MTBM = Mean Time Between Maintenance*
*MDT = Mean maintenance Down Time*

EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**Term: Safety**
DEF: 1. Safety S(t) is the probability that a system will either perform its functions correctly or will discontinue its functions in a manner that does not disrupt the operation of other systems or compromise the safety of any people associated with the system. Safety is a measure of the *fail-safe* capability of a system; if the system does not operate correctly, you at least want the system to fail in a safe manner. For example, a pilot can safely fly an airplane, even if the autopilot fails, as long as the failure does not inhibit the aircraft's normal flight modes. Likewise, if a control valve for a chemical process fails, you often prefer that the valve will fail in the closed position. Safety is the probability that these safe actions will result.
SOURCE: 1. Barry W. Johnson, Design and Analysis of Fault Tolerant Digital Systems, Addison-Wesley Publishing Company, 1985, p. 6.
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**Term: Performability**
DEF: 1. The performability P(L,t) of a system is a function of time, defined as the probability that the system performance will be at, or above, some level L, at the instant of time t [Fortes and Raghavendra 1984]. If we relate performability to the multiprocessor example, the level of performance might simply be the number of processors available for computational use. Performability differs from reliability in that reliability is a measure of the likelihood that *all* of the functions are performed correctly, whereas performability is a measure of the likelihood that some subset of the functions is performed correctly.
SOURCE: 1. Barry W. Johnson, Design and Analysis of Fault Tolerant Digital Systems, Addison-Wesley Publishing Company, 1985, p. 6.
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**Term: Maintainability**
DEF: 1. Maintainability is a measure of the ease with which a system can be repaired once it has failed. In more quantitative terms, maintainability M(t) is the probability that a failed system will be restored to an operational state within a specified period of time t. The restoration process includes locating the problem, physically repairing the system, and bringing the system back to its operational condition. Maintainability is crucial in all systems, but it is particularly

| | |
|---|---|
| | important when human lives, equipment, or the environment are placed in jeopardy while a system is repaired. |
| SOURCE: | 1. Barry W. Johnson, Design and Analysis of Fault Tolerant Digital Systems, Addison-Wesley Publishing Company, 1985, p. 7. |
| METRICS: | 1. N/A |
| EXAMPLE: | 1. N/A |
| USAGE: | 1. N/A |
| NOTE: | 1. N/A |

**TERM: Ease of Replacement**

| | |
|---|---|
| DEF: | 1. The ease of replacement is the amount of time, money, effort, etc., that is necessary to replace the system. It generally assumes a replacement system is available. |
| SOURCE: | 1. NSWCDD, CJW |
| METRICS: | 1. System dependent, but generally will include an amount of time and/or money, etc. |
| EXAMPLE: | 1. It will take an installation crew of two people 10 hours to install a new system at a cost of $20,000. |
| USAGE: | 1. N/A |
| NOTE: | 1. N/A |

**TERM: Crash Recoverability**

| | |
|---|---|
| DEF: | 1. A description of the process necessary to regain some, if not full, system operation after various types of system "crashes." |
| SOURCE: | 1. NSWCDD, CJW |
| METRICS: | 1. N/A |
| EXAMPLE: | 1. N/A |
| USAGE: | 1. N/A |
| NOTE: | 1. N/A |

**TERM: Computation Heavy Process Effects**

| | |
|---|---|
| DEF: | 1. A description of how availability of the system is effected under unusually "stressful" or "loaded" situations. |
| SOURCE: | 1. NSWCDD, CJW |
| METRICS: | 1. System dependent |
| EXAMPLE: | 1. When busy compiling missile threat information, all other processes on the system will run 50 percent slower. |
| USAGE: | 1. N/A |
| NOTE: | 1. N/A |

**TERM: Quality**

| | |
|---|---|
| DEF: | 1. This gauge developed by JPL laboratories in Pasadena, California, is meant to be a relative measure of the number of errors in a piece of software. |
| SOURCE: | 1. Bush, M. W., "Getting Started on Metrics - Jet Propulsion Laboratory Productivity and Quality," Proceedings from the 12th International Conference on Software Engineering, 1990, p.134. |
| METRICS: | 1. QUALITY = defects per thousand lines of source code (DEF/KSLOC) |
| EXAMPLE: | 1. The study from which this measure was developed found that their own large scale flight system software had an average quality of 8.6 defects per thousand lines of code. Their ground systems software had an average quality of 2.1. |
| USAGE: | 1. N/A |
| NOTE: | 1. N/A |

# SECURITY FACTOR

**TERM: Security**

| | |
|---|---|
| DEF: | 1. The degree to which the system can protect its contents and operation from unauthorized use. |
| SOURCE: | 1. NSWCDD, CJW |
| METRICS: | 1. N/A |
| EXAMPLE: | 1. N/A |
| USAGE: | 1. N/A |
| NOTE: | 1. N/A |

**TERM: (Security) Level**

| | |
|---|---|
| DEF: | 1. Security may need to meet a particular standard, SECRET or TOP SECRET, for example. |
| SOURCE: | 1. NSWCDD, CJW |
| METRICS: | 1. N/A |
| EXAMPLE: | 1. N/A |
| USAGE: | 1. N/A |
| NOTE: | 1. N/A |

# HUMANWARE FACTOR

**TERM: Humanware**

DEF: 1. Characteristics describing the interfaces between the system and the rest of the world, i.e., humans.
SOURCE: 1. NSWCDD, CJW
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Ease of Use**

DEF: 1. How "user friendly" the system is, including background or training necessary for a user to operate the system.
SOURCE: 1. NSWCDD, CJW
METRICS: 1. System dependent.
EXAMPLE: 1. Ease of use can run the spectrum from an on-line tutorial to teach the user to turn the system on and aid the user from any point, to a 6 week training course to teach the user to operate the system.
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Potential Operator Decisions**

DEF: 1. A description of any important or critical decision a system operator may need to make.
SOURCE: 1. NSWCDD, CJW
METRICS: 1. System dependent.
EXAMPLE: 1. In a nuclear cruise missile launch system, the user would probably make the final decision to fire the missile.
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: 1. Operator Delay/2. User Response Time**

DEF: 1. A list of any possible delays caused by the user critical to the system's operation.
2. Time needed by a user at an interactive terminal to generate a new request (think and type time).
SOURCE: 1. NSWCDD, CJW
2. Svobodova, Liba, Computer Performance Measurement and Evaluation Methods: Analysis and Applications, 1976, p.13.
METRICS: 1. System dependent.
EXAMPLE: 1. A keyboard input can expect characters only as fast as a human can type (on the order of 100 words per minute).
2. Referring to the example under POTENTIAL OPERATOR DECISIONS, the user that decides to tell the system to launch the missile may need to first consult a superior officer which could typically take so many minutes or seconds.
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Operator Action(s)**

DEF: 1. A description of important and significant actions a user must make when operating the system.
SOURCE: 1. NSWCDD, CJW
METRICS: 1. System dependent.
EXAMPLE: 1. For most large, complex systems, the total number of possible actions a user could take are probably enormous. The utility of this design factor may be more applicable to a fully automated system where, for example, the only actions an operator can take are to turn the system on and off.
USAGE: 1. N/A

NOTE: 1. N/A

**TERM: 1. Required Number of Operators/2. Number of Simultaneous Users**

DEF: 1. The number of operators/users that are needed for system operation under various conditions.
2. Number of interactive users logged on concurrently.

SOURCE: 1. NSWCDD, CJW
2. Svobodova, Liba, Computer Performance Measurement and Evaluation Methods: Analysis and Applications, 1976, p.13.

METRICS: 1. System dependent, although this metric typically will be just a number of human beings (i.e., operators/users), needed to operate the system under various conditions.

EXAMPLE: 1. For normal operation, a navigation system may only need two operators and one commander, while during a wartime situation the system may need four operators and one commander.

USAGE: 1. N/A

NOTE: 1. N/A

**TERM: User Intensity**

DEF: 1. A measure of "how much" work the system user is doing compared to "how much" work the system itself is doing.

SOURCE: 1. Svobodova, Liba, Computer Performance Measurement and Evaluation Methods: Analysis and Applications, 1976, p.13.

METRICS: 1. (Processing time per request) ÷ (user response time)

EXAMPLE: 1. The system is processing each request every 10 seconds with the user producing a new request every 20 seconds. This equates to a user intensity of .5.

USAGE: 1. N/A

NOTE: 1. N/A

# PHYSICAL REQUIREMENTS FACTOR

**TERM: Physical Requirements**
DEF: 1. Descriptions on the actual material, mechanical form that the system is to take.
SOURCE: 1. NSWCDD, CJW
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Size Requirements**
DEF: 1. Characterization of the amount of space that the system can occupy.
SOURCE: 1. NSWCDD, CJW
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Height**
DEF: 1. Limits on the vertical length of the system.
SOURCE: 1. NSWCDD, CJW
METRICS: 1. An appropriate length measurement; meters, feet, inches, for example.
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Area**
DEF: 1. Limits on the floor space the system can occupy.
SOURCE: 1. NSWCDD, CJW
METRICS: 1. An appropriate area measurement, for example, square meters, square feet, square inches.
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Volume**
DEF: 1. Limits on the cubic extent the system can occupy.
SOURCE: 1. NSWCDD, CJW
METRICS: 1. An appropriate volume measurement, for example, cubic meters, liters, cubic feet.
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Weight Requirements**
DEF: 1. NSWCDD, CJW
SOURCE: 1. Limits on the expected mass of the system.
METRICS: 1. An appropriate mass measurement, for example, kilograms, pounds, slugs.
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Survivability**
DEF: 1. A description of how much of varying types of physical abuse the system can take (from an enemy for example), before it is disabled or destroyed.
SOURCE: 1. NSWCDD, CJW
METRICS: 1. System dependent.
EXAMPLE: 1. The system must be bullet proof and must not be damaged from a 10-foot hard drop.

USAGE: 1. N/A
NOTE: 1. N/A

**TERM: (Physical) Portability**
DEF: 1. Degree to which, or with how much ease or difficulty, a system can be moved from location to location.
SOURCE: 1. NSWCDD, CJW(dict)
METRICS: 1. System dependent.
EXAMPLE: 1. System can easily be held and carried by one person and only has a standard U.S. electrical power cord to be connected or disconnected from its operating environment.
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Energy Requirements**
DEF: 1. Description of the type, quality, and quantity of energy needed by the system for operation and dissipated by the system during operation.
SOURCE: 1. NSWCDD, CJW
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: (Energy) Consumption**
DEF: 1. A listing of the resources consumed by the system during operation.
SOURCE: 1. NSWCDD, CJW
METRICS: 1. System dependent, although energy consumed by the system will generally be measured by some sort of composite of measures of the factors listed under this catagory.
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Electrical (Energy Consumed)**
DEF: 1. This will be a description of the electrical needs of the system, including any standards, that the electrical supply must meet.
SOURCE: 1. NSWCDD, CJW
METRICS: 1. AC or DC supply, volts, amps, power (watts), phase (Hz).
EXAMPLE: 1. The system will need an electrical supply of 110V AC, 5 amps at 60 Hz. This can be supplied by any standard American electrical wall outlet.
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Other (Energy Consumed)**
DEF: 1. A description of any non-electrical energy needs of the system.
SOURCE: 1. NSWCDD, CJW
METRICS: 1. System dependent.
EXAMPLE: 1. The solar cells powering the system will need at least X hrs of Y lumens intensity sunlight a day for proper operation.
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: (Energy) Dissipated**
DEF: 1. A listing of the resources dissipated by the system during operation.
SOURCE: 1. NSWCDD, CJW
METRICS: 1. System dependent, although energy dissipated by the system will generally be measured by some sort of composite of measures of the dissipations produced by the system.

EXAMPLE: 1. The system's electronics produce 10 watts of heat during operation.
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Locational Operating Environment**
DEF: 1. A description of the environment and climate where the system will be placed and expected to be operational.
SOURCE: 1. NSWCDD, CJW
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. The subcatagories that fall under this factor are designed to be all inclusive and as such contain a great deal of interrelationships and dependencies. For example, if a system is listed in the INDOORS/OUTDOORS factor as always being contained indoors, it is most likely the case that the system will be listed as experiencing no water exposure under the EXPOSURE TO WATER factor.
2. Parts of the system, or the entire system itself may need to have climate modifications as described in the CLIMATE CONTROL section. The LOCATIONAL OPERATING ENVIRONMENT section is an actual description of the absolute environment where the system is located.

**TERM: Geographical Location**
DEF: 1. Where on the globe the system is expected to be deployed.
SOURCE: 1. NSWCDD, CJW
METRICS: 1. System dependent.
EXAMPLE: 1. Latitude and Longitude; geographical region such as artic, desert, mountains, etc.; a specific country or continent; etc.
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Indoors/Outdoors**
DEF: 1. If the system is to be located within an enclosed structure (indoors), or exposed to the elements (outdoors), or in some cases both.
SOURCE: 1. NSWCDD, CJW
METRICS: 1. Indoors, outdoors, or both indoors and outdoors.
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Temperature**
DEF: 1. N/A
SOURCE: 1. NSWCDD, CJW
METRICS: 1. A range of expected temperatures in degrees Celsius, Fahrenheit, or Kelvin.
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Humidity**
DEF: 1. N/A
SOURCE: 1. NSWCDD, CJW
METRICS: 1. A range of expected humidity stated as either absolute water content of the air or as a percentage equal to the water content of the air relative to the maximum possible water content of the air at the ambient temperature (relative humidity).
EXAMPLE: 1. N/A
USAGE: 1. N/A

NOTE: 1. N/A

**TERM: Acoustical Noise**
DEF: 1. N/A
SOURCE: 1. NSWCDD, CJW
METRICS: 1. System dependent, but sound is typically measured in decibles of sound pressure level.
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Air Purity/Quality**
DEF: 1. N/A
SOURCE: 1. NSWCDD, CJW
METRICS: 1. System dependent.
EXAMPLE: 1. The air to which the system will be exposed will contain airborn contaminants and impurities larger than 100 microns on the scale of 100 parts per million.
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Exposure to Wind**
DEF: 1. N/A
SOURCE: 1. NSWCDD, CJW
METRICS: 1. System dependent.
EXAMPLE: 1. The outdoor environment in which the system will be deployed typically experiences winds of 10 mph on average with gusts up to 50 mph.
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Exposure to Water**
DEF: 1. N/A
SOURCE: 1. NSWCDD, CJW
METRICS: 1. System dependent.
EXAMPLE: 1. The outdoor environment in which the system will be deployed experiences water exposure typically no greater than the equivalent of light rain.
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Exposure to Electromagnetic Radiation**
DEF: 1. N/A
SOURCE: 1. NSWCDD, CJW
METRICS: 1. System dependent, although the standard measurement for E/M radiation is watts per square meter at stated frequencies.
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Vibrations/Stability**
DEF: 1. N/A
SOURCE: 1. NSWCDD, CJW
METRICS: 1. System dependent.
EXAMPLE: 1. Due to its in-vehicle deployment, the system can expect shocks and vibrations in all directions equivalent to a drop of 3 feet onto a hard surface.
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Climate Control**
DEF: 1. Aspects of the environment surrounding the system that the system itself will need to control.
SOURCE: 1. NSWCDD, CJW
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Cooling**
DEF: 1. N/A
SOURCE: 1. NSWCDD, CJW
METRICS: 1. System dependent, although typically this will include a range of temperatures that a cooling system would be expected to maintain.
EXAMPLE: 1. The computer facility's air conditioner must maintain an ambient air temperature of no more than 70° F.
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Heating**
DEF: 1. N/A
SOURCE: 1. NSWCDD, CJW
METRICS: 1. System dependent, although this will typically include a range of temperatures that a heating system would be expected to maintain.
EXAMPLE: 1. The computer facility's heating system must maintain an ambient air temperature of at least 60° F.
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Humidity Control**
DEF: 1. N/A
SOURCE: 1. NSWCDD, CJW
METRICS: 1. System dependent, although this will typically include a range of humidities that a humidification/dehumidification system would be expected to maintain.
EXAMPLE: 1. The computer facility's humidification/dehumidification system must maintain an ambient relative humidity of between 40 percent and 70 percent.
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Acoustical Noise Suppression**
DEF: 1. N/A
SOURCE: 1. NSWCDD, CJW
METRICS: 1. System dependent, but typically will be a decibel level of reduction in the amount of sound aborbed by, or emanated from, the system.
EXAMPLE: 1. Surrounding the system in a 3-inch layer of fiberglass insulation will provide the 10dB reduction in acoustical noise necessary to render the system effectively silent.
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Air Purity/Quality Control**
DEF: 1. N/A
SOURCE: 1. NSWCDD, CJW
METRICS: 1. System dependent.
EXAMPLE: 1. All airborne contaminants larger than 100 microns must be filtered out of the air surrounding the system.

USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Motion Stabilization**
DEF: 1. N/A
SOURCE: 1. NSWCDD, CJW
METRICS: 1. System dependent.
EXAMPLE: 1. All components of the system will need to be securely fastened with enough strength to hold against an impulse force of up to 500 lbs in any direction.
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Lighting**
DEF: 1. N/A
SOURCE: 1. NSWCDD, CJW
METRICS: 1. System dependent.
EXAMPLE: 1. The user's operating area will need lighting typical of normal indoor working areas. It is estimated that four 48 inch fluorescent light fixtures will accommodate this need.
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Manufacturing Considerations**
DEF: 1. Qualities, characteristics, and requirements on the actual physical production of a system.
SOURCE: 1. NSWCDD, CJW
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Production Capacity**
DEF: 1. The maximum number of systems that can be produced in a given time.
SOURCE: 1. NSWCDD, CJW
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Production Time**
DEF: 1. The amount of time necessary to produce/construct a system.
SOURCE: 1. NSWCDD, CJW
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Life Cycle Costs**
DEF: 1. Costing refers to a formula for weighing and qualifying the total cost of a system from the beginning of the design to the final disposal of the equipment. It is a conceptually appealing because it brings to the forefront all of the factors that are affected by design decisions, which in isolation seem innocuous.
2.
SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 86-87).
2. NSWCDD, CMN
METRICS: 1. N/A

2. Money unit, X amount of dollars per year for Y number of years

EXAMPLE: 1. In the following, we shall deal with costs from two points of view: first the slit between engineering (nonrecurring) and manufacturing (recurring) costs and second, a more general view of life cycle cost of a piece of equipment. In some situations, it is possible to partition the overall activities into two parts: engineering and manufacturing. For our initial purposes, engineering implies everything involved in the design and production of the first prototype. And manufacturing is the activity associated with creating a product from the engineering prototype and then building and delivering it. The general problem here is to armotize the two activities over the expected number of products:

$$\text{cost unit} = \frac{\text{engineering costs}}{\text{number of units}} + \frac{\text{fixed manufacturing costs}}{\text{number of units}} + \text{variable manufacturing cost(per unit)}$$

A life cycle cost is computed by considering the costs associated with all phases of the equipment (regardless of who pays them). The total life of the equipment may be divided into

a. design,
b. development of prototypes,
c. manufacturing,
d. operation and maintenance, and
e. disposal or replacement.

2. The cost to produce system A would be on the average X millions of dollars per year for Y years

USAGE: 1. N/A

2. Life cycle costs are used to estimate the funds required to develop and maintain a system or a project. These costs are used for proposals, budgeting, feasibility studies, and design decisions.

NOTE: 1. N/A

# FINANCIAL REQUIREMENTS FACTOR

**TERM: Financial Requirements**

DEF: 1. Restrictions and expectations on the money involved in varying aspects of the system.
SOURCE: 1. NSWCDD, CJW
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Cost to Develop**

DEF: 1. The financial amount expended on labor and overhead to produce a version of the system that passes formal qualification tests.

SOURCE: 1. CCCC, EL
2. Boehm, B. W., Software Engineering Economics, Prentice-Hall, Englewood Cliffs, NJ 1981.
3. Arthur, L. J., Measuring Programmer Productivity and Software Quality, Wiley-Interscience, New York, 1985.
4. Dreger, J.B., Function Point Analysis, Prentice Hall, Englewood Cliffs, NJ, 1989.

METRICS: 1. Dollars (or any unit of currency); amount or percentage deviation from planned costs.
Opportunity Cost--i.e., what else could be done with the resources.
Material cost.
Labor cost.
Present value.
$/SLOC.

EXAMPLE: 1. There are many techniques for estimating the cost to develop. Some of these are listed below:
a. Pressman: Make 3 estimates, most likely (m), optimistic (o), and pessimistic (p). The formula for the expected estimate (e) is $e = (o + 4m+p)/6$.
b. COnstructive COst MOdel (COCOMO): This model was developed by Barry Boehm. It relies upon equations of the form E=a*(L**b) where E is the effort (or time) and L is the number of lines of code. The model focusses on producing families of (a,b) values to account for project specific factors.
c. Function Points: This model estimates a project's cost as a function of the target system's attributes rather than its predicted size. Typically, five system properties are used to compute the function point count (FP). They are the number of user inputs, number of user outputs, number of user inquiries, number of files, and number of external interfaces.
FP is the weighted sum of these five counts. [Arth] provides weighting factors for simple, average, and complex systems. A typical formula would be:
FP = 4*inputs + 5*outputs + 4*inquiries + 10*files + 7*interfaces.
The number of function points per line of code varies per language--on average, 100 to 120 lines of COBOL code or 60-65 lines of PL/1 code for each function point produced. For 4GLs, there are typically only 15 lines per function point.
The average $/SLOC for Ada systems is between $60 and $65 per SLOC.

USAGE: 1. Useful for planning and management of software projects. In planning, development cost is needed to perform cost benefit analysis. During development, the cost to develop is important for gauging progress and determining the project's financial needs.

NOTE: 1. The term formal qualification test comes from DOD-STD-2167A and is used to refer to the final test that a development effort must pass.
SLOC is source lines of code.

**TERM: Cost to Prototype**

DEF: 1. The amount of resources required to produce a version of the system that will help validate the user's requirements.
SOURCE: 1. CCCC,EL
METRICS: 1. Same metrics as cost to develop. Size of prototype is less than size of total system and thus will be less costly.
EXAMPLE: 1. N/A

USAGE: 1. Perform rapid prototyping to reduce the risk associated with not understanding the user's requirements or the user not knowing what is needed.
NOTE: 1. N/A

**TERM: Cost to Produce**
DEF: 1. All costs required to manufacture additional copies of the system.
SOURCE: 1. CCCC,EL
METRICS: 1. $/unit (where unit is a copy).
Total cost.
EXAMPLE: 1. The development might amount to millions of dollars, but the cost to produce additional copies might amount to less than $20 to copy it onto some media (floppy, tape, CD, etc.) and ship.
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Cost to Test**
DEF: 1. The amount of resources required to take a discrete item of software and determine if it satisfies a set of requirements.
SOURCE: 1. CCCC,EL
METRICS: 1. See COST TO DEVELOP
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. There are different types of testing, such as unit test, integration test, inspection, review, etc.

**TERM: Cost to Purchase**
DEF: 1. The amount of resources needed to acquire ownership rights or other license rights.
SOURCE: 1. CCCC,EL
METRICS: 1. See COST TO DEVELOP
EXAMPLE: 1. Cost may also include time required to make purchasing decision, adapt product, and train in the product.
USAGE: 1. Typically purchase costs imply that the product or service already existed or the responsibility for making it exist lies within the vendor of the good or service.
NOTE: 1. N/A

**TERM: Cost to Operate**
DEF: 1. Amount of resources required to use the system.
SOURCE: 1. CCCC,EL
METRICS: 1. See COST TO DEVELOP, training cost
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Cost to Maintain**
DEF: 1. Amount of resources required to perfect, adapt, or correct a system.
SOURCE: 1. CCCC,EL
METRICS: 1. See COST TO DEVELOP
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Cost to Repair**
DEF: 1. The amount of resources required to take a system that does not satisfy user requirements and make it satisfy user requirements.
SOURCE: 1. CCCC,EL
METRICS: 1. See COST TO DEVELOP

EXAMPLE: 1. $/defect, total cost, number of lives saved/lost
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Cost to Include Security Capability**
DEF: 1. The amount of resources required to make a completely non-secure system secure.
SOURCE: 1. CCCC,EL
METRICS: 1. See COST TO DEVELOP
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Productivity**
DEF: 1. This set of measures both labeled productivity were developed by JPL laboratories in Pasadena, California. They are meant to give a relative measure of how much software is being produced for the amount of money and labor involved.
2. Outputs produced by the process divided by inputs consumed by the process.
SOURCE: 1. Bush, M. W., "Getting Started on Metrics - Jet Propulsion Laboratory Productivity and Quality," Proceedings of the 12th International Conference on Software Engineering, 1990, p.134.
2. Boehm, B., "Improving Software Productivity," IEEE Computer, September 1987, pp. 43-57.
METRICS: 1. PRODUCTIVITY = source lines of code per work month (SLOC/WM)
PRODUCTIVITY = dollars per source line of code ($/SLOC)
2. Outputs may include Delivered Source Instructions (DSI), function points, control flow metrics, and work transaction metrics. Inputs may include unit of time, unit of exchange, phases (e.g., software development), activities (e.g., documentation, project management), personnel, resources (e.g., facilities, equipment).
EXAMPLE: 1. The study from which these measures were developed found that their own large scale flight system software had an average productivity of 10 source lines of code per work month and $1,149 per line of source code. Their ground systems software had an average productivity of 186 SLOC/work month and $67 per SLOC.
2. SLOC per hour
USAGE: 1. N/A
2. DSI's may be counted with and without comments or blank lines, or executable statements. SLOC is source lines of code.
NOTE: 1. N/A
2. Refer to Boehm's article for life cycle productivity ranges.

# TIME PROJECTED FACTOR

# THIS CHAPTER HAS NOT YET BEEN DEFINED

# LIFE CYCLE FACTOR

**TERM: Life Cycle**

DEF: 1. The complete process of bringing a system into being that starts with the identification of a need and extends through planning, research, design, production or construction, evaluation, consumer use, maintenance and support, and ultimately retirement (phaseout).
2. A description of the expected life of the system throughout all of its phases.
SOURCE: 1. Blanchard and Fabrycky, Systems Engineering and Analysis, 1990, p.17.
2. NSWCDD, CJW.
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Testability**

DEF: 1. The ability to evaluate conformance with requirements
SOURCE: 1. Nance, R. E., and Arthur, J. D., "Developing an Automated Procedure for Evaluating Software Development Methodologies and Associated Products," Virginia Polytechnic Institute, Systems Research Center, Technical Report SRC-87-007, April 16, 1987.
METRICS: 1. OPA Framework (see Technical Report)
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Maintainability**

DEF: 1. The ease with which corrections can be made to respond to recognized inadequacies.
2. How difficult it will be to correct errors found in the field.
SOURCE: 1. Nance, R. E.,and Arthur, J. D., "Developing an Automated Procedure for Evaluating Software Development Methodologies and Associated Products," Virginia Polytechnic Institute, Systems Research Center, Technical Report SRC-87-007, April 16, 1987.
2. Shooman, M. L., "Software Engineering," McGraw-Hill, New York, 1983.
METRICS: 1. OPA Framework (see Technical Report)
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: (Maintenance) Notification**

DEF: 1. How the user will be signaled when maintenance is necessary. More importantly, whether or not the system itself will keep track of when maintenance is necessary.
SOURCE: 1. NSWCDD, CJW.
METRICS: 1. N/A
EXAMPLE: 1. A "maintenance" idiot light scheduled by an internal system clock, a written pencil, and paper log to be kept by the user, etc.
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: (Maintenance) Frequency**

DEF: 1. How often scheduled maintenance should be performed.
SOURCE: 1. NSWCDD, CJW.
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Maintenance Downtime/(Maintenance) Duration**

DEF: 1. The total elapsed time required (when the system is not operational) to repair and restore a system to full operating status.

| | |
|---|---|
| | 2. The amount of time necessary for maintenance. |
| SOURCE: | 1. Blanchard and Fabrycky, Systems Engineering and Analysis, 1990, p.403. |
| METRICS: | 1. MDT - measured in days, hours, minutes, etc. |
| EXAMPLE: | 1. N/A |
| USAGE: | 1. N/A |
| NOTE: | 1. N/A |

**TERM: Degree of System Disability: 1. When Maintenance Comes Due**
**2. During Maintenance**

| | |
|---|---|
| DEF: | 1. A description of the degree of functionality the system will have when maintenance becomes due.<br>2. A description of the degree of functionality the system will have during maintenance. |
| SOURCE: | 1. NSWCDD, CJW |
| METRICS: | 1. N/A |
| EXAMPLE: | 1. One extreme would be that the system may shut down when maintenance is due, and the other extreme would be that performing maintenance on the system may have no operational effect on the system whatsoever. |
| USAGE: | 1. N/A |
| NOTE: | 1. N/A |

**TERM: Maintainer**

| | |
|---|---|
| DEF: | 1. A description of who will perform what maintenance. |
| SOURCE: | 1. NSWCDD, CJW |
| METRICS: | 1. N/A |
| EXAMPLE: | 1. The user may be able to perform the necessary maintenance, or it may need to be a specific repair person, or contracted company. |
| EXAMPLE: | 1. N/A |
| USAGE: | 1. N/A |
| NOTE: | 1. N/A |

**TERM: Wear Lifetime**

| | |
|---|---|
| DEF: | 1. The expected or estimated life of the system. When a "worn out" system may be defined to need updating, maintenance, overhaul, or complete replacement. |
| SOURCE: | 1. NSWCDD, CJW |
| METRICS: | 1. N/A |
| EXAMPLE: | 1. N/A |
| USAGE: | 1. N/A |
| NOTE: | 1. N/A |

**TERM: Obsolescence lifetime**

| | |
|---|---|
| DEF: | 1. The period of time for the functionality of software to become invalid.<br>2. The expected or estimated time before the system's technology and capabilities become unable to perform the tasks demanded of them. **Caution:** This is a characteristic that will inevitably involve a great deal of speculation due to the very subjective nature of the term "obsolescence." |
| SOURCE: | 1. CCCC,EL<br>2. NSWCDD, CJW |
| METRICS: | 1. unit of time (years, months, days, hours, minutes, seconds, etc.)<br>unit of operation (after you use it x times) |
| EXAMPLE: | 1. N/A |
| USAGE: | 1. N/A |
| NOTE: | 1. N/A |

**TERM: Reusability**

| | |
|---|---|
| DEF: | 1. The use of developed software and/or its associated documentation in other applications. |

SOURCE: 1. Nance, R. E.,and Arthur, J. D., "Developing an Automated Procedure for Evaluating Software Development Methodologies and Associated Products," Virginia Polytechnic Institute, Systems Research Center, Technical Report SRC-87-007, April 16, 1987.
METRICS: 1. OPA Framework (see Technical Report)
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Correctness**
DEF: 1. Strict adherence to specified requirements.
2. The consistency of a product with respect to its specifications.
SOURCE: 1. Nance, R. E.,and Arthur, J. D., "Developing an Automated Procedure for Evaluating Software Development Methodologies and Associated Products," Virginia Polytechnic Institute, Systems Research Center, Technical Report SRC-87-007, April 16, 1987.
2. Blum, B.I., "Software Engineering: A Holistic View," Oxford University Press, pp. 363-367, 470-471, 1992.
METRICS: 1. OPA Framework (see Technical Report)
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Reliability**
DEF: 1. The error free use of software over time.
2. The probability of an undiscovered defect. Mean Time to Repair (MTTR) is an estimate of the time to code a solution.
SOURCE: 1. Nance, R. E.,and Arthur, J. D., "Developing an Automated Procedure for Evaluating Software Development Methodologies and Associated Products," Virginia Polytechnic Institute, Systems Research Center, Technical Report SRC-87-007, April 16, 1987.
2. Blum, B.I., "Software Engineering: A Holistic View, Oxford University Press, pp. 363-367, 470-471, 1992.
METRICS: 1. OPA Framework (see Technical Report)
2. Probability distribution
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Portability**
DEF: 1. The ease in transferring software to another environment.
2. In terms of Ada, the number of non-portable constructs used within a program.
SOURCE: 1. Nance, R. E., and Arthur, J. D., "Developing an Automated Procedure for Evaluating Software Development Methodologies and Associated Products," Virginia Polytechnic Institute, Systems Research Center, Technical Report SRC-87-007, April 16, 1987.
2. CCCC,EL.
METRICS: 1. OPA Framework (see Technical Report)
2. The number of non-portable constructs used or percentage of lines that are non-portable
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Adaptability**
DEF: 1. The ease with which software can accommodate to change.
SOURCE: 1. Nance, R. E., and Arthur, J. D., "Developing an Automated Procedure for Evaluating Software Development Methodologies and Associated Products," Virginia Polytechnic Institute, Systems Research Center, Technical Report SRC-87-007, April 16, 1987.
METRICS: 1. OPA Framework (see Technical Report)

| | |
|---|---|
| **EXAMPLE:** | **1. N/A** |
| **USAGE:** | **1. N/A** |
| **NOTE:** | **1. N/A** |

# FUTURE NEEDS CONSIDERATIONS FACTOR

**TERM: Future Needs Considerations**

DEF: 1. Specifications addressing the potential for the system to take on requirements or abilities not presently intended to be imposed on the system.
SOURCE: 1. NSWCDD, CJW
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Adaptability/Flexibility**

DEF: 1. The extent to which the system can take on new tasks and needs (new and updated software for a specific example), without the need for any significant modifications.
SOURCE: 1. NSWCDD, CJW
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Expandability**

DEF: 1. A measure of the degree and ease with which "things" can be added or modified on the system to allow for changes in the system's capabilities.
SOURCE: 1. NSWCDD, CJW
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Compatibility**

DEF: 1. A description of other systems, peripherals, communication links, etc. that can be used with the system.
SOURCE: 1. NSWCDD, CJW.
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Integrity**

DEF: 1. The attributes of integrity are the attributes that limit the number of false starts or inappropriate considerations that must be made. It is also related to the number of iterations between design steps required to focus on a suitable candidate.
SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 68).
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Complexity**

DEF: 1. The issue of complexity may be thought to have been contained within the scope of the attribute. However, the issue is not the inherent complexity of the problem (this is part of the scope of the methodology) but within the intrinsic presence, in the methodology itself, of complexity not attributable to the original specifications. The methodology should be complex only to the extent demanded by the complexity inherent in the original system. A good methodology, therefore, must constrain (i.e., limit) complexity. It should also perceive and expose simple patterns and relationships throughout the design cycle.

| | |
|---|---|
| SOURCE: | 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 68). |
| METRICS: | 1. N/A |
| EXAMPLE: | 1. N/A |
| USAGE: | 1. N/A |
| NOTE: | 1. N/A |

## REFERENCES

[BlF90] Blanchard and Fabrycky, Systems Engineering and Analysis, 1990.

[BoB85] Bowen, B. A., and Brown, W. R., System Design: Volume II of System Design for Digital Signal Processing, Prentice-Hall, Inc., 1985.

[CYH91] Choi, D.; Youngblood, J.; Hwang, P., "Modeling Technology for Dynamic Systems," Proc. 1991 Systems Evaluation and Assessment Technology Workshop, Aug 1991.

[Dos91] Doskocil, D., "Modeling Techniques to Support Diagnostics System Trade-off," Proc. 1991 Systems Evaluation and Assessment Technology Workshop, Aug 1991.

[EdH91] Edwards, M. and Howell, S., "Requirements Specification and Traceability (RESPECT): A Requirements Methods for Large Complex Systems," Proc. 1991 Systems Design Synthesis Technology Workshop, Sep 1991.

[HNH92] Howell, S.; Nguyen, C.; Hwang, P., "Design Structuring and Allocation Optimization (DeStinAtiOn): A Front-end Methodology for Prototyping Large, Complex, Real-Time Systems," Proc. Hawaii International Conference on System Sciences, IEEE Computer Society Press, Los Alamistos, CA, Jan 1992, Vol. II, pp 517-528.

[HNH91] Howell, S.; Nguyen, C.; Hwang, P., "System Design Structuring and Allocation Optimization (DeStinAtiOn)," Proc. 1991 Systems Design Synthesis Technology Workshop, Sep 1991.

[HHN90a] Howell, S.; Hwang, P.; Nguyen, C., "Expert Design Advisor," Proc. 5th Jerusalem Conference on Information Technology (JCIT), IEEE Computer Society Press, Los Alamistos, CA, Oct 1990, pp 743-756.

[HHN90b] Howell, S.; Hwang, P.; Nguyen, C., "Expert Design Advisor," Naval Surface Warfare Center Technical Report, TR90-46, Oct 1990.

[Hoa91] Hoang, ND. T., "Essential Views of Systems Development," Proc. 1991 Systems Design Synthesis Technology Workshop, Sep 1991.

[JeY91] Jenkins, M. and Yeh, C., "An Approach to Design of Processor Networks Based On Massively Interconnected Models," Proc. 1991 Systems Design Synthesis Technology Workshop, Sep 1991.

[Joh85] Johnson, B. W., Design and Analysis of Fault Tolerant Digital Systems, Addison-Wesley Publishing Company, 1985.

[JuA91] Juttelsatd, D. and Arnold, C., "The Next Generation Computer Resources (NGCR) Operating System Metrics Project," Proc. 1991 Systems Evaluation and Assessment Technology Workshop, Aug 1991.

[Kam91] Kamat, V., "Computer System Evaluation: Paths and Pitfalls," Proc. 1991 Systems Evaluation and Assessment Technology Workshop, Aug 1991.

[NaF91] Mansour N., and Fox, G., "Physical Optimization Methods for Allocating Data to Multicomputer Nodes," Proc. 1991 Systems Design Synthesis Technology Workshop, Sep 1991.

[SvL76] Svobodova, and Liba, Computer Performance Measurement and Evaluation Methods: Analysis and Applications, 1976.

[WaH91] Wallenberger, P. and Howell, S., "Real-Time Dependable Systems Design," Proc. 1991 Systems Design Synthesis Technology Workshop, Sep 1991.

# APPENDIX A

## LIST OF DEFINITIONS

This appendix provides definitions of terms that are commonly used by the systems engineering community. For each term a definition(s) is provided for a better understanding, a source(s) for references, metrics to provide the unit of measurement, example(s) to provide an application, usage(s) to provide guidance and, finally, note(s) to provide additional comments.

**TERM:**
DEF: 1. N/A
SOURCE: 1. Bowen, B. A., and, Brown W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 95).
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Bottlenecks**
DEF: 1. Often, several potentially useful logical descriptions will emerge, and criteria are required to form a list of candidates for further consideration. Each may be subjected to scrutiny by bottlenecks. It may be possible from the logical structure to identify bottlenecks in performance. For example, if the requirements contain data-flow rates and response times, it should be possible to relate these constraints to the logical structure. Such an examination will often suggest alternative partitions that alleviate potential bottlenecks.
SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 96).
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. Refer to terms Implementation Dependent and Logical Complexity.

**TERM: Bottom-Up Design**
DEF: 1. This approach begins with an existing set of the system components, or at least a subset of components, and extends and modifies them to meet the requirements. Mature design shops often pursue this strategy, since they usually integrated, perhaps with some patches, to from the required system. The bottom-up strategy is philosophically opposite to a top-down approach. The approach is clearly one of successive compositions (as opposed to refinements). The composition terminates at the top with a system that fulfills the specifications. There is no doubt that this has proven successful and economical for many systems. Provided the ultimate extension to the final system is within reach of the existing inventory of basic modules, the approach can be made to work. Pathologies exist in two way: First, it is often difficult to predict if the desired system can be implemented exactly, and second, it is equally difficult to predict the cost. A fine tuned sense of realizability is a key attribute of a successful design venture starting with large inventory of existing components.
SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 59).
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Constraint-Driven Design**
DEF: 1. The reality of most design environments (indeed, the distinguished features of an engineering design) is the existence of constraints. The function of design in this environment is to satisfy (constraints) not necessary to optimize. An approach which accommodates all design worth realities of the critical components and constraints is called constraint-driven. This approach has two important advantages: (a) convergence on an acceptable solution (from top or bottom) is more likely to occur with fewer iterations; and (b) the impact of a constraint is accommodated at an appropriate level in the overall strategy.
SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 62-63).
METRICS: 1. N/A
EXAMPLE: 1. N/A

USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Control-Driven Tactics**
DEF: 1. Control-driven tactics tend to be more universally familiar. The identifications to be carried out and their precedence and control features appear as a natural approach for real-time processing and process control problems.
SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 64).
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Data-Driven Tactics**
DEF: 1. Data-driven design tactics recognize the flow of data as the driving force in the system design and attempt to identify the processing function that supports the data flow structure first and then defines the necessary control structures.
SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 65).
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Design**
DEF: 1. (noun) A set of plan or sketches, and patterns or a model, or some other form of description of something to be implemented or executed.
2. (verb, transitive) To work out, to indicate, to plan mutually, to outline, to fashion according to plan, to picture, to sketch as a pattern or model, to execute as a whole.
3. (verb, intransitive) To conceive or execute a scheme or plan.
SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 53).
2. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 53).
3. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 53).
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Design Mechanism**
DEF: 1. Design mechanism is a tool which is intended to facilitate a particular aspect of either the strategic or tactical portion of the overall design process. Design mechanisms range from the representation of the design process itself to notational conventions such as access graphs, structure diagrams, flow charts, decision tables, pseudo-codes, etc.
SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 56).
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Design Methodology**

DEF: 1. A specific study of the principles and procedures for creating a design.
SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 53).
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Design Strategy**
DEF: 1. Design strategy is a prescribed overall sequence (of steps) or a general approach by which major components of the design are created.
SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 55).
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Design Tactic**
DEF: 1. A specific procedure for implementing a step or more steps in the overall strategy.
2. A basic similarity in all of the strategies is the attempt to modularize the overall system design, i.e., (a) successive refinement of major system modules from the top down; (b) successive composition of major modules from the bottom up; and (c) design of the most critical system modules first, etc.
SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 55).
1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 64).
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Edges-In/Interfaces First**
DEF: 1. Given a system requirements specification this methodology suggests that the designer first separate the non-functional requirements and constraints, then define the required system functions beginning with the system interfaces of the "edges" of the system. This is accomplished by trying several different functional partitions and drawing block diagrams of the data-flow and control-flow requirements for each partition.
SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 71).
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Execution Time Budgets**
DEF: 1. Once the data-flow rates have been calculated for each edge of the data-flow graph, these data rates are combined with the data element consumption and production definitions for each node to calculate minimum node execution time budgets. These represent the minimum execution time for the node such that no bottleneck in system data flow occurs at the node.
SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 130).
METRICS: 1. N/A
EXAMPLE: 1. N/A

USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Horizontal Partitioning**
DEF: 1. A horizontal partitioning in which there is no data-flow across the partitions divides the system into non-interacting data-flow paths. This type of system partitioning can lead to systems composed of autonomous parallel units that do not require close synchronization, and which can, therefore, be easily implemented on parallel processors.
SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 133).
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Horizontal-Vertical and Vertical-Horizontal Partitioning**
DEF: 1. A complex network data-flow organization can usually be partitioned in stages, with either an initial overall vertical partition and subsequent horizontal subpartitions or a primary horizontal partitioning with vertical subpartitions. The various approaches also impact the expandability and fail soft features of the system, maintenance considerations, spacing, MTTR,etc.
SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 133).
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Hybrids Strategies**
DEF: 1. Hybrid strategies employed most critical components first, outside-in, inside-out, and edges-in/interfaces first design approaches.
SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 95).
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Implementation**
DEF: 1. The steps involve the myriad details of selecting the hardware components and actually creating the hardware and the software which, in combination, become the required system.
SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 55).
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Implementation Dependency**
DEF: 1. Often, several potentially useful logical descriptions will emerge, and criteria are required to form a list of candidates for further consideration. Each may be subjected to scrutiny by implementation dependency. Strictly speaking, the design at this level should be independent of implementation details. Often, however, previous experience will suggest attractive ways to proceed. Indeed, in some cases a bottom-up design can be initiated with a probability of success. In general, these candidates should be tagged and others explored before a

commitment is made. Specifically, it is desirable to avoid premature commitments to hardware or software or software implementation. The factors affecting the optimum selection of the boundary between the implementation mechanisms are changing rapidly.

SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 96).

METRICS: 1. N/A

EXAMPLE: 1. N/A

USAGE: 1. N/A

NOTE: 1. Refer to Logical Complexity and Bottlenecks.

**TERM: Logical Complexity**

DEF: 1. Often, several potentially useful logical descriptions will emerge, and criteria are required to form a list of candidates for further consideration. Each may be subjected to scrutiny by Logical Complexity. Simplicity is desirable at this stage (and many others as well). The logical description of the functions that are overly complex have three pathologies. First, the complexity will probably increase during implementation; second, it may indicate that the operational requirements were not well understood; and third, it may inhibit a critical evaluation and validation.

SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 96).

METRICS: 1. N/A

EXAMPLE: 1. N/A

USAGE: 1. Refer to Implementation Dependent and Bottlenecks.

NOTE: 1. N/A

**TERM: Logical Description**

DEF: 1. The logical description translates the system requirements into data flow and control elements. These are iteratively decomposed to a level of detail sufficient to partition and allocate them to specific hardware of specific software for implementation and on, essentially, a one-to-one basis.

SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 89-90).

METRICS: 1. N/A

EXAMPLE: 1. N/A

USAGE: 1. N/A

NOTE: 1. It is important to note that a theoretical description of the concepts of the required processing is not an algorithm description, although such a description may be necessary as part of the derivation of the algorithm to be implemented. The algorithm description itself must define, at some appropriate level of detail, exactly what the logical data inputs, transformation, and outputs are for the required processing system in terms of machine-implementable functions, operations, and data representations. It is the specific mechanisms that allow such descriptions to be formulated that represent the fundamental tools of the systems designer.

**TERM: Logical Design**

DEF: 1. The result of the logical design activity is a complete description of what is to be implemented. It is, therefore, a critical phase of the methodology. While the designer is ultimately concerned with performance, the first concern here is with logical correctness. From the set of logically correct alternatives, the prime candidates are those exhibiting an elegance, marked by simplicity, in both data flow and control. While hard to quantify and measure, the solution should only be as complex as is intrinsically required by the problem. Logical design process begins with a set of requirements and organizes these into a logical description of the data-flow and control features. This description then forms the basis for selecting a physical system implementation. The logical design procedure is intended to create a description of the system that defines what the system does, and how it is logically organized, to a level of detail such that this description can be mapped to some hardware architecture for implementation.

SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 89-91,96).
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. Refer to conceptual model, architecture model, implementation design, or implementation description for counter part.

**TERM: Logical Level**
DEF: 1. What the system must do.
SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 95).
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Logical System Design**
DEF: 1. This phase involves the interpretation of the specifications and the creation of a set of virtual machines that can ultimately be realized as some combination of hardware and software.
SOURCE: 1. Bowen, B. A., and Brown W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 54).
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: MIL-STD-490 System Specification**
DEF: 1. The availability of a standard specification format can be very useful as a guideline, both for preparing specifications and for analyzing them. While there are no universally accepted standards for specification formats, standards do exist. MIL-STD-490 is the general format or structural outline that applies to all types and serves as a useful guideline for ensuring that all the necessary information is contained in a specification.
SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 88-89).
METRICS: 1.
EXAMPLE: 1. MIL-STD-490 System Specification
- a. Scope
- b. Applicable documents
- c. Requirements
  - System definition
    - General description
    - Application Environments
    - Interfaces
    - Major system elements
    - Operation description
  - Characteristics
    - Performance
    - Physical
    - Reliability
    - Maintainability
    - Availability
    - Systems effectiveness models (acceptance criteria)
    - Operational environment conditions
  - Documentation

Maintenance
Supply
Support facilities and equipment
- Personnel and training
Functional characteristics

d. Quality assurance
- General
- Responsibility for test
- Special test and examination
- Quality conformal and inspections

e. Preparations for delivery

f. Notes/appendices

USAGE: 1. The structure forms a useful guideline for either formulating specifications or for examining a proposed set of specifications.
NOTE: 1. N/A

**TERM: Networks Data-Flow**
DEF: 1. Combinations of both sequential and parallel data flows can lead to networks, which may take an unlimited number of more complex forms. A key issue in logical system design is the partitioning of the data-flow functions to provide maximum data-flow flexibility while attempting to minimize network control complexity.
SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 133).
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Nonquantifiable Constraints**
DEF: 1. In almost every set of specifications, there is a range of attributes that is difficult, if not impossible, to quantify. In most cases, these constraints are not measurable as the direct result of the widely accepted model but agreed upon for the duration of the design project.
SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 86).
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Orthogonal Strategies**
DEF: 1. Orthogonal strategies employe top-down and bottom-up design approaches.
SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 55).
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Outside-In/Inside-Out Approaches**
DEF: 1. The approaches are often designated as strategies but are really variations of the top-down approach that recognizes different tops. The outside of a system is thought of as the part the user sees, i.e., the user interface. Generally, there is a distinct difference between the user's perception of what the system is doing and what is actually occurring (i.e., he or she se a victual machine). A system may be thought of as two logical blocks, the first being system algorithm as seen by the designer and the second, the user interface that creates his/her

perspective. Thus, if the internal operational decisions are made first, followed by the design of the user interface, the approach is inside-out. Conversely, if decisions are made first relative to the user interface, then the approach is outside-in.

SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 61).
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Parallel Data-Flow**
DEF: 1. Several data streams flow through several functions in parallel, with no interaction of precedence relationships existing between the separate data streams.
SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 133).
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Partitioning Consideration**
DEF: 1. The data flow analysis will lead to some initial partitioning and allocation ideas for the system. The data-flow graphs show the inheritance maximum for the required computations as well as the inherent precedence relationships. The representation of data-flow graph requirements as data graphs can lead to a variety of data flow organizations, which can be used to guide the system partitioning. The data-flow organization can be classified as sequential, parallel, or network (combination of sequential and parallel).
SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 132).
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Partitioning Control Flow**
DEF: 1. The interconnection topology of multiple-processor systems affects the system performance in that it introduces communication and synchronization overhead. Therefore, partitions must attempt to minimize this overhead, usually by control flow consideration. The control flow partition boundaries are located so as to minimize the synchronization requirements between modules.
SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 98).
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Partitioning Data Flow**
DEF: 1. The interconnection topology of multiple-processor systems affects the system performance in that it introduces communication and synchronization overhead. Therefore, partitions must attempt to minimize this overhead, usually by data flow consideration. The data flow partitioning boundaries are located across the structured diagram which has minimum data flow traffic , i.e., a minimum bandwidth requirement. These boundaries are often found at the conclusion of major data-flow activities.

SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 98).
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Partitioning with Respect to Data-Flow**
DEF: 1. As the major data-flow requirements of the systems are represented, first as the single nodes of data-flow graph and then successively refined, the overall structure of the data flow graph can be used as a basis for system partitioning. If the data-flow graphs were drawn such that the data always flowed from left to right across the page, then the partitioning graph can be consider either with vertical lines or with horizontal lines.
SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 133).
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Physical Description**
DEF: 1. Physical description of the system can be given in terms of its hardware and software components.
SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 89).
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Physical Level**
DEF: 1. How the system is physically implemented.
SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 95).
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Quantifiable Constraints**
DEF: 1. Quantifiable constraints in the specification are those that can be computed or predicted from an analysis of the system and its components. This does not imply that the mechanism for doing so is straightforward or, in some cases, even well understood. Quantifiable constraints lead to acceptance tests that must eventually be passed if the system is to be accepted by the customer. It is essential that their influence on the design decisions be understood; more important, perhaps, is to determine what decisions made during the design process affect these constraints. It is difficult to formulate general guidelines. However, in specific cases, each quantifiable constraint should be explicitly considered at every decision point in the design.
SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 84-85).
METRICS: 1. N/A
EXAMPLE: 1. N/A
USAGE: 1. N/A
NOTE: 1. N/A

**TERM: Sequential Data-Flow**

DEF: 1. Data flow sequentially from one function to the next.

SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 133).

METRICS: 1. N/A

EXAMPLE: 1. N/A

USAGE: 1. N/A

NOTE: 1. N/A

**TERM: Specification**

DEF: 1. This activity involves the creation of the document from the user requirements that reflect the nature of the problem to be solved in a way that drives the design and that also serves to provide quantifiable validations of the results of the design. Thus, the specification serves as the beginning and the end of the design process.

SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 54).

METRICS: 1. N/A

EXAMPLE: 1. N/A

USAGE: 1. N/A

NOTE: 1. N/A

**TERM: Structural Attributes of Design Methodologies**

DEF: 1. The various structural attributes of a design methodology are considered in terms of design strategies, design tactics, and design mechanisms. Each methodology has a basic structure that in turn might be thought of as having two parts. The first is the sequence of steps (i.e., the strategies), with an associated set of principles that are used to guide decisions in areas of uncertainty (e.g., insufficient quantifiable data). The second part, corresponding to a tactical procedure for carrying out the strategy, is concerned with making design decisions based on the input data (and/or the information derived from those data) that must be made and accommodated at prescribed points in the design process. The tactics employed at each step rely heavily on the specific mechanisms employed.

SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 56, 68).

METRICS: 1. N/A

EXAMPLE: 1. N/A

USAGE: 1. N/A

NOTE: 1. N/A

**TERM: System Functions and Initial Partitioning**

DEF: 1. Given the system requirements specification, the designer's first task is to identify the major functional requirements of the system and partition these into the initial set of systems level and virtual machines based on the user visibility of the system functions and operational characteristics associated with each. In general, these system functions can be broken down into broad categories of data flow and control functions.

SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 91).

METRICS: 1. N/A

EXAMPLE: 1. N/A

USAGE: 1. N/A

NOTE: 1. N/A

**TERM: Systems Design**

DEF: 1. The final system design is a detailed definition of both the hardware and software and how they operate together to carry out the required processing. The overall system consists of the logical description, physical description, and the tangle mapping between the two. The process of system design is concerned with all aspects of moving from some statement of requirements and constraints to a final definition of the hardware and software that meet the stated requirements within the specified constraints

SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 89).

METRICS: 1. N/A

EXAMPLE: 1. N/A

USAGE: 1. N/A

NOTE: 1. Refer to conceptual model, architecture model, logical design, logical description, implementation design, or implementation description for counter part.

**TERM: The Most-Critical-Components-First Approach**

DEF: 1. This approach is often included as a design strategy by various authors. In it, those parts of the system whose operation is most constrained are designed first. This approach is the variation of both top-down and bottom-up approaches. From the top-down point of view, it assures the designer that the critical operational constraints can be met. From the bottom-up, an assurance is also obtained that critical systems functions can be executed. The approach implies that some knowledge has been obtained already as to the logical structure of the system in which the critical components fit. It is proposed that this approach is not a design strategy but rather a criterion for partitioning the logical functions of the system and also a constraint on allocation of those functions to either hardware or software.

SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 62).

METRICS: 1. N/A

EXAMPLE: 1. N/A

USAGE: 1. N/A

NOTE: 1. N/A

**TERM: Top-Down Design**

DEF: 1. Top-down designs are widely advocated for software projects and are often proposed for all system designs. This approach performs a progression of tasks that successively define layers of the system. The top-down approach is axiomatic in that the precedence relation exist between the layers that it can be found. Such a procedure guarantees that larger questions are answered before smaller ones and that the logical structure is determined before the system is embedded in the concrete of details. Because of this feature, it is often referred to as a successive refinement approach. The benefits of pursuing this strategic approach to design are correctness, clarity, and modularity. The successive layering of the system from abstract requirements to implementation provides advantages in the understanding and, hence, in communicating the design to others. This also materially assists in guaranteeing correctness as each level evolves. The various layers of the system represent a horizontal partitioning that builds an inherent modularity into the design. Within the layers, further vertical partitions can enhance this modularity. This forces the designer to consider control requirements and interface definitions at each stage of the design.

SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 57-59).

METRICS: 1. N/A

EXAMPLE: 1. N/A

USAGE: 1. N/A

NOTE: 1. N/A

**TERM: Vertical Partitioning**

DEF: 1. A vertical partition divides the system into functions with definite sequential precedence relationship in a pipe line fashion corresponding to an overall sequential data-flow organization. This type of partitioning can lead to a logical structure corresponding to a collection of functional units operating in parallel with data sets sequentially passed from one unit to the next.

SOURCE: 1. Bowen, B. A., and Brown, W. R., Systems Design: Volume II of Systems Design for Digital Signal Processing, Prentice-Hall, Inc., 1985 (pp 133).

METRICS: 1. N/A

EXAMPLE: 1. N/A

USAGE: 1. N/A

NOTE: 1. N/A

# APPENDIX B

## WORKING GROUP MEMBERS AND POINTS OF CONTACT

A list of System Design Factors Working Group members follows. The list includes the coordinator's and members' organization, phone, fax, email, and volunteer responsibility.

NAME: Dong Choi
COMPANY: Naval Surface Warfare Center
ADDRESS: Naval Surface Warfare Center
Attn: Dong Choi, Code B44
10901 New Hampshire Ave.
Silver Spring, MD 20903-5640
PHONE: (301)394-1913
FAX: (301)394-1175
E-MAIL: dchoi@relay.nswc.navy.mil

NAME: Edgar Cohen
COMPANY: Naval Surface Warfare Center
ADDRESS: Naval Surface Warfare Center
Attn: Edgar Cohen, Code B44
10901 New Hampshire Ave.
Silver Spring, MD 20903-5640
PHONE: (301)394-2281
FAX: (301)394-3923
E-MAIL: ecohen@nswc-wo.navy.mil

NAME: Michael Edwards
COMPANY: Naval Surface Warfare Center
ADDRESS: Naval Surface Warfare Center
Attn: Michael Edwards, Code B44
10901 New Hampshire Avenue
Silver Spring, MD 20903-5640
PHONE: (301) 394-4187
FAX: (301) 394-1175
E-MAIL: medward@relay.navy.mil

NAME: Geoffry Frank
COMPANY: Research Triangle Institute
ADDRESS: Research Triangle Institute
Attn: Geoffry Frank
Ctr for Digital Systems Research
Research Triangle Park, NC 27709
PHONE: (919)541-6629
FAX:
E-MAIL: gaf@rti.rti.org

NAME: Robert T. Goettge
COMPANY: Advanced System Technologies, Inc.
ADDRESS: Advanced System Technologies, Inc.
Attn: Robert Goettge
12200 E. Briarwood Ave., Suite 260
Englewood, CO 80112
PHONE: (303)790-4242
FAX: (303)790-2816
E-MAIL:

NAME: Ngocdung Hoang
COMPANY: Naval Surface Warfare Center
ADDRESS: Naval Surface Warfare Center
Attn: Ngocdung Hoang, Code B44
10901 New Hampshire Avenue
Silver Spring, MD 20903-5640
PHONE: (301) 394-4877
FAX: (301) 394-1175
E-MAIL: nhoang@relay.nswc.navy.mil

NAME: Steven L. Howell
COMPANY: Naval Surface Warfare Center
ADDRESS: Naval Surface Warfare Center
Attn: Steven L. Howell, Code B44
10901 New Hampshire Ave.
Silver Spring, MD 20903-5640
PHONE: (301)394-3987
FAX: (301)394-1175
E-MAIL: showell@relay.nswc.navy.mil

NAME: Michael Jenkins
COMPANY: Naval Surface Warfare Center
ADDRESS: Naval Surface Warfare Center
Attn: Michael Jenkins, Code B44
10901 New Hampshire Ave.
Silver Spring, MD 20903-5640
PHONE: (301) 394-4139
FAX: (301) 394-1175
E-MAIL: mjjenki@relay.nswc.navy.mil

NAME: Nicholas E. Karangelen
COMPANY: Trident Systems Incorporated
ADDRESS: Trident Systems Incorporated
10201 Lee Hwy, Suite 300
Fairfax, VA 22030
PHONE: (703)273-1012
FAX: (703)273-6608
E-MAIL:

NAME: Jane W. S. Liu
COMPANY: University of Illinois at Urbana-Champaign
ADDRESS: U. of Illinois at Urbana-Champaign
Attn: Jane W. S. Liu
Department of Computer Science
1304 W. Springfield Avenue
Urbana, IL 61801
PHONE: (217)333-0135
FAX:
E-MAIL: janeliu@cs.uiuc.edu

NAME: Evan Lock
COMPANY: Computer Command Control & Company
ADDRESS: CCCC
Attn: Evan Lock
2300 Chestnut Street, Suite 230
Philadelphia, PA 19103
PHONE: (215)854-0555
FAX: (215)854-0665
E-MAIL: lock@cccc.com

NAME: Jogesh K. Muppala
COMPANY: Duke University
ADDRESS: Duke University
Attn: Kishor Trivedi
Dept. of Electrical Eng.
Durham, NC 27706
PHONE:
FAX:
E-MAIL:

NAME: Cuong M. Nguyen
COMPANY: Naval Surface Warfare Center
ADDRESS: Naval Surface Warfare Center
Attn: Cuong M. Nguyen, Code B44
10901 New Hampshire Ave.
Silver Spring, MD 20903-5640
PHONE: (301)394-4550
FAX: (301)394-3179
E-MAIL: cnguyen@relay.nswc.navy.mil

NAME: Eric Ogata
COMPANY: Naval Surface Warfare Center
ADDRESS: Naval Surface Warfare Center
Attn: Eric Ogata, Code G42
10901 New Hampshire Ave.
Silver Spring, MD 20903-5640
PHONE: (301)394-2355
FAX: (301)394-1175
E-MAIL: ogata@degas.nswc.navy.mil

NAME: Carl Schmiedekamp
COMPANY: Naval Air Warfare Center
ADDRESS: Naval Air Warfare Center
Attn: Carl Schmiedekamp
Code 7033
Warminster, PA 18974-5000
PHONE: (215) 441-1779
FAX: (215) 441-3225
E-MAIL: carls@nadc.navy.mil

NAME: Harold Szu
COMPANY: Naval Surface Warfare Center
ADDRESS: Naval Surface Warfare Center
Attn: Harold Szu, Code B44
10901 New Hampshire Ave.
Silver Spring, MD 20903-5640
PHONE: (301)394-3097
FAX: (301)394-3923
E-MAIL: hszu@relay.nswc.navy.mil

NAME: My-Hanh N. Trinh
COMPANY: Naval Surface Warfare Center
ADDRESS: Naval Surface Warfare Center
Attn: My-Hanh Trinh, Code B44
10901 New Hampshire Ave.
Silver Spring, MD 20903-5640
PHONE: (301)394-5372
FAX: (301)394-1175
E-MAIL: mtrinh@relay.nswc.navy.mil

# INDEX

# DISTRIBUTION

| | Copies |
|---|---|
| DoD Activity (CONUS) | |
| Defense Technical Information Center<br>Cameron Station<br>Alexandria, VA 22304-6145 | 12 |
| Non-DoD Activity | |
| Library of Congress<br>Attn: Gift & Exchange Division<br>Washington, DC 20540 | 4 |
| DoD Activities (CONUS) | |
| Naval Air Warfare Center<br>Attn: Code 7033 | |
| (Dr. Carl Schmiedekamp) | 1 |
| (Dr. Jessy Williams) | 1 |
| (Dr. Ming Ming Chen) | 1 |
| Warminster, PA 18974-5000 | |
| Naval Postgraduate School | |
| Attn: (Dr. Chin-Hwa Lee) | 1 |
| (Dr. Norman Schneidewind) | 1 |
| Dept. of Electrical & Computer Engineering<br>Monterey, CA 93943 | |
| Naval Research Laboratory<br>Attn: Code 5583 | |
| (Roger Hillson) | 1 |
| (Kathy Meadow) | 1 |
| 4555 Overlook Ave., SW<br>Washington, D.C. 20375-5000 | |
| Naval Undersea Warfare Center<br>Attn: Code 2151 | |
| (Dr. Thomas Choinsky) | 1 |
| (Dan J. Organ) | 1 |
| New London, CT 06320 | |
| Naval Ocean Systems Center | |
| Attn: Dr. Keith Bromley | 1 |
| Space Systems & Technology Division<br>Code 7601 T<br>271 Catalina Blvd.<br>San Diego, CA 92152-5000 | |
| Office of Naval Research<br>Attn: Code 227A | |
| (Elizabeth Wald) | 1 |
| (CDR. Gracie L. Thompson) | 1 |
| 800 N. Quincy Street<br>Arlington, VA 22217-5660 | |

| | Copies |
|---|---|
| Non-DoD Activities | |
| Advanced System Technologies, Inc. | |
| Attn: (Eric Brehm) | 1 |
| (Dr. Robert Goettge) | 1 |
| (Goerge Krasovic) | 1 |
| (Chuck Palczak) | 1 |
| (Dr. Rayner K. Rosich) | 1 |
| 12200 E. Briarwood Ave., Suite 260<br>Englewood, CO 80112 | |
| Advanced System Technologies, Inc. | |
| Attn: William Dudzik | 1 |
| 5113 Leesburg Pike, Suite 514<br>Falls Church, VA 22041 | |
| Allied-Signal Aerospace Company | |
| Attn: (Dr. Jeffrey X. Zhou) | 1 |
| (Dr. Michelle McElvany Hugue) | 1 |
| (Dr. Dar-Tzen Peng) | 1 |
| Aerospace Technology Center<br>9140 Old Annapolis Road/MD 108<br>Columbia, MD 21045-1998 | |
| Automated Sciences Group, Inc. | |
| Attn: Rajiv Jain | 1 |
| 1781 Dahlgren Road<br>Dahlgren, VA 22448 | |
| Carnegie Mellon University | |
| Attn: Dr. Jay Strosnider | 1 |
| Dept. of Electrical & Computer Engineering<br>Pittsburgh, PA 15213 | |
| Center for Naval Analyses<br>4401 Fort Avenue<br>P.O. Box 16268 | |
| Alexandria, VA 22302-0268 | 2 |
| Computer Command and Control Company | |
| Attn: (Scott W. Andrews) | 1 |
| (Dr. Jee-In Kim) | 1 |
| (Dr. Insup Lee) | 1 |
| (Dr. Wu-Hung Liu) | 1 |
| (Evan Lock) | 1 |
| (Baba Prasad) | 1 |
| (Rajesh Purushothaman) | 1 |
| (Dr. Noah Prywes) | 1 |
| 2300 Chestnut Street, Suite 230<br>Philadelphia, PA 19103 | |

| | Copies |
|---|---|
| Conceptual Software Systems, Inc.<br>Attn: Ed Andert, Jr.<br>P.O. Box 727<br>Yorba Linda, CA 92686 | 1 |
| Data & Analysis Center for Software<br>Attn: James J. Reed<br>P. O. Box 120<br>Utica, NY 13503 | 1 |
| Decision Science Associates, Inc.<br>Attn: Jim Chinnis<br>10980 Polar Ford Trail<br>Manassas, VA 22110-2106 | 1 |
| Duke University<br>Attn: Dr. Kishor Trivedi<br>Dept. of Electrical Eng.<br>Durham, NC 27706 | 1 |
| General Dynamics, Space Systems Division<br>Attn: Jeffery Grady<br>6015 Charae St.<br>San Diego, CA 92122 | 1 |
| General Electric Company<br>Attn: David W. Oliver<br>Bldg. K-1, Room 3C3<br>P.O. Box 8<br>Schenectady, NY 12301 | 1 |
| George Mason University<br>Attn: Richard Evans<br>ST11 RM 133<br>4400 University Dr.<br>Fairfax, VA 22030-4444 | 1 |
| Grumman CRC<br>Attn: Dr. Stephanie White<br>MS A08-35<br>Bethpage, NY 11714-3580 | 1 |
| IBM Corp.<br>Attn: Will Tracz<br>Route 17C, Mail Drop 0210<br>Owego, NY 13827 | 1 |
| IBM T.J. Watson Research Center<br>Attn: Dr. Farnam Jarhanian<br>P.O. Box 704<br>Yorktown Heights, NY 10598 | 1 |

| | Copies |
|---|---|
| JHU/APL<br>Attn: Dr. Bruce Blum<br>Johns Hopkins Rd.<br>Laurel, MD 20723-6099 | 1 |
| Mitre Corp.<br>Attn: William Evanco<br>M/S Z385<br>7525 Colshire Drive<br>McLean, VA 22102-3481 | 1 |
| New Jersey Institute of Technology | |
| Attn: (Carlos C. Amaro) | 1 |
| (Raja S. Daita) | 1 |
| (Michael Khamis) | 1 |
| (Dr. Phillip Laplante) | 1 |
| (Dr. Thomas Marlowe) | 1 |
| (Dr. Alex D. Stoyenko) | 1 |
| (Dr. Lonnie R. Welch) | 1 |
| (Jin Xue) | 1 |
| Computer and Information Sciences Department<br>University Heights<br>Newark, NJ 07102 | |
| Research Triangle Institute<br>Attn: Dr. Geoffry Frank<br>Ctr for Digital Systems Research<br>Research Triangle Park, NC 27709 | 1 |
| Science and Technology Associates | |
| Attn: (Mary E. Blanchard) | 1 |
| (Gary S. Schebella) | 1 |
| 4001 N. Fairfax Dr., Suite 700<br>Arlington, VA 22203 | |
| Texas A & M University | |
| Attn: (Dr. Wei Zhao) | 1 |
| (Dr. Swaminathan Natarajan) | 1 |
| Department of Computer Science<br>College Station, TX 77843-3112 | |
| Trident Systems Incorporated | |
| Attn: (David Brittan) | 1 |
| (Michael Cassey) | 1 |
| (Nicholas E. Karangelen) | 1 |
| (Glen Wood) | 1 |
| (Shane Bailey) | 1 |
| 10201 Lee Hwy, Suite 300<br>Fairfax, VA 22030 | |

## DISTRIBUTION

| | Copies |
|---|---|
| DoD Activity (CONUS) | |
| Defense Technical Information Center | |
| Cameron Station | |
| Alexandria, VA 22304-6145 | 12 |
| Non-DoD Activity | |
| Library of Congress | |
| Attn: Gift & Exchange Division | 4 |
| Washington, DC 20540 | |
| DoD Activities (CONUS) | |
| Naval Air Warfare Center | |
| Attn: Code 7033 | |
| (Dr. Carl Schmiedekamp) | 1 |
| (Dr. Jessy Williams) | 1 |
| (Dr. Ming Ming Chen) | 1 |
| Warminster, PA 18974-5000 | |
| Naval Postgraduate School | |
| Attn: (Dr. Chin-Hwa Lee) | 1 |
| (Dr. Norman Schneidewind) | 1 |
| Dept. of Electrical & Computer Engineering | |
| Monterey, CA 93943 | |
| Naval Research Laboratory | |
| Attn: Code 5583 | |
| (Roger Hillson) | 1 |
| (Kathy Meadow) | 1 |
| 4555 Overlook Ave., SW | |
| Washington, D.C. 20375-5000 | |
| Naval Undersea Warfare Center | |
| Attn: Code 2151 | |
| (Dr. Thomas Choinsky) | 1 |
| (Dan J. Organ) | 1 |
| New London, CT 06320 | |
| Naval Ocean Systems Center | |
| Attn: Dr. Keith Bromley | 1 |
| Space Systems & Technology Division | |
| Code 7601 T | |
| 271 Catalina Blvd. | |
| San Diego, CA 92152-5000 | |
| Office of Naval Research | |
| Attn: Code 227A | |
| (Elizabeth Wald) | 1 |
| (CDR. Gracie L. Thompson) | 1 |
| 800 N. Quincy Street | |
| Arlington, VA 22217-5660 | |

| | Copies |
|---|---|
| Non-DoD Activities | |
| Advanced System Technologies, Inc. | |
| Attn: (Eric Brehm) | 1 |
| (Dr. Robert Goettge) | 1 |
| (Goerge Krasovic) | 1 |
| (Chuck Palczak) | 1 |
| (Dr. Rayner K. Rosich) | 1 |
| 12200 E. Briarwood Ave., Suite 260 | |
| Englewood, CO 80112 | |
| Advanced System Technologies, Inc. | |
| Attn: William Dudzik | 1 |
| 5113 Leesburg Pike, Suite 514 | |
| Falls Church, VA 22041 | |
| Allied-Signal Aerospace Company | |
| Attn: (Dr. Jeffrey X. Zhou) | 1 |
| (Dr. Michelle McElvany Hugue) | 1 |
| (Dr. Dar-Tzen Peng) | 1 |
| Aerospace Technology Center | |
| 9140 Old Annapolis Road/MD 108 | |
| Columbia, MD 21045-1998 | |
| Automated Sciences Group, Inc. | |
| Attn: Rajiv Jain | 1 |
| 1781 Dahlgren Road | |
| Dahlgren, VA 22448 | |
| Carnegie Mellon University | |
| Attn: Dr. Jay Strosnider | 1 |
| Dept. of Electrical & Computer Engineering | |
| Pittsburgh, PA 15213 | |
| Center for Naval Analyses | |
| 4401 Fort Avenue | |
| P.O. Box 16268 | |
| Alexandria, VA 22302-0268 | 2 |
| Computer Command and Control Company | |
| Attn: (Scott W. Andrews) | 1 |
| (Dr. Jee-In Kim) | 1 |
| (Dr. Insup Lee) | 1 |
| (Dr. Wu-Hung Liu) | 1 |
| (Evan Lock) | 1 |
| (Baba Prasad) | 1 |
| (Rajesh Purushothaman) | 1 |
| (Dr. Noah Prywes) | 1 |
| 2300 Chestnut Street, Suite 230 | |
| Philadelphia, PA 19103 | |

| | Copies |
|---|---|
| UniView Systems<br>Attn: Dr. Armen Gabrielian<br>1192 Elena Privada<br>Mountain View, CA 94040 | 1 |
| Virginia Polytechnic Institute & State University<br>Attn: (Dr. James D. Arthur) | 1 |
| (Dr. Osman Balci) | 1 |
| (Dr. Richard Nance) | 1 |
| (Dinesh Verma)<br>Department of Computer Science<br>Blacksburg, VA 24061 | 1 |
| University of California<br>Attn: (Dr. K. H. (Kane) Kim) | 1 |
| (Dr. Kwei-Jay Lin)<br>Dept. of Electrical & Computer Engineering<br>Irvine, CA 92717 | 1 |
| University of Illinois at Urbana-Champaign<br>Attn: (Zhong Deng) | 1 |
| (Dr. Jane W. S. Liu) | 1 |
| (Too-Seng Tia)<br>Department of Computer Science<br>1304 W. Springfield Avenue<br>Urbana, IL 61801 | 1 |
| University of Michigan<br>Attn: Dr. Kang G. Shin<br>Dept. of Electrical & Computer Science<br>1301 Beal Avenue<br>Ann Arbor, MI 48109-2122 | 1 |
| Univesity of York<br>Attn: Dr. Andy Welling<br>Dept. of Computer Science<br>York YO1 5DD<br>United Kingdom | 1 |
| University of Wisconsin<br>Attn: Dr. Parameswaran Ramanathan<br>Dept. of Electrical & Computer Engineering<br>1415 Johnson Drive<br>Madison, WI 53706-1691 | 1 |

| | Copies |
|---|---|
| University of Kansas<br>Attn: Dr. Julian Holtzman<br>CECASE<br>2291 Irving Hill Road<br>Nicholas Hall - W. Campus<br>Lawrence, KS 66045 | 1 |

| | Copies | Copies |
|---|---|---|
| Internal Distribution: | | |
| A10 (R Scalzo) | 1 | |
| B | 1 | |
| B02 | 1 | |
| B04 | 1 | |
| B05 (H. Crisp) | 20 | |
| B10 | 1 | |
| B10 (D. Parks) | 1 | |
| B10 (W. Farr) | 1 | |
| B10 (H. Huber) | 1 | |
| B20 | 1 | |
| B30 | 1 | |
| B40 | 1 | |
| B42 (J. Moscar) | 1 | |
| B44 (E. Cohen) | 1 | |
| B44 (T. Dumoulin) | 1 | |
| B44 (D. Dorsey) | 1 | |
| B44 (M. Edwards) | 1 | |
| B44 (N. Hoang) | 1 | |
| B44 (S. Howell) | 20 | |
| B44 (M. Jenkins) | 1 | |
| B44 (C. Nguyen) | 20 | |
| B44 (H. Roth) | 1 | |
| B44 (H. Szu) | 1 | |
| B44 (M. Trinh) | 1 | |
| B44 (J. Wingate) | 1 | |
| B44 (M. Wilson) | 1 | |
| B44 (J. Zien) | 1 | |
| D4 | 1 | |
| E231 | 2 | |
| E232 | 3 | |
| E342 (GIDEP) | 1 | |
| F01 | 1 | |
| F31 (W. Laposata) | 1 | |
| G07 (F. Moore) | | |
| 1 | | |
| G72 (H. Parks) | 1 | |
| H32 (J. Miller) | 1 | |
| K02 | 1 | |
| K10 (J. Sloop) | 1 | |
| K14 (D. Clark) | 1 | |
| K41 (L. Gross) | 1 | |
| K51 (J. Smith) | 1 | |
| K52 (G. Brooks) | 1 | |
| N35 (M. Masters) | 1 | |
| N35 (F. Riedl) | 1 | |
| N35 (M. Zarin) | 1 | |

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704-0188*



**1. AGENCY USE ONLY** *(Leave blank)*

**2. REPORT DATE**
18 March 1994

**3. REPORT TYPE AND DATES COVERED**
Final

**4. TITLE AND SUBTITLE**
Systems Design Factors: The Essential Ingredients of System Design Version 0.4

**5. FUNDING NUMBERS**

**6. AUTHOR(S)**
Cuong M. Nguyen and Steven L. Howell

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**
Naval Surface Warfare Center
Dahlgren Division, White Oak Detachment (Code B40)
10901 New Hampshire Avenue
Silver Spring, MD 20903-5640

**8. PERFORMING ORGANIZATION REPORT NUMBER**
NSWCDD/TR-92/268

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

**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

**11. SUPPLEMENTARY NOTES**

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

**12b. DISTRIBUTION CODE**

**13. ABSTRACT** *(Maximum 200 words)*
The key to designing a real-time, large, complex system is to optimize the design to meet the requirements and desired measure of effectiveness. In order to achieve this, the system engineer/analyst must have the capability to specify the design goals/criteria, to quantify various aspects of the design, and to perform trade-offs among different design goals. One of the mechanisms that provides these capabilities is the System Design Factors. Whether the system design emphasis is on real-time, largeness, complexity, parallelism or any specific criteria, it requires a set of System Design Factors to describe the properties, attributes and characteristics of the system. Each System Design Factor must have its own metric to gauge every detail of that system. The metric describes the weaknesses and strengths of a specific area in the design. In turn, the correlation of the System Design Factor characterizes the completeness and robustness of the system. Whether the system is designed top-down, bottom-up, or middle-out, the System Design Factors have major influence in design capture and analysis, design structuring decisions, allocation decisions, and trade-off decisions between various design structures and resource allocation candidates.

**14. SUBJECT TERMS**
Systems Design Factors, Structure Design, Resource Allocation, Optimization, Trade-off Analysis, Large Complex Real-Time System

**15. NUMBER OF PAGES**
102

**16. PRICE CODE**

| 17. SECURITY CLASSIFICATION OF REPORT | 18. SECURITY CLASSIFICATION OF THIS PAGE | 19. SECURITY CLASSIFICATION OF ABSTRACT | 20. LIMITATION OF ABSTRACT |
|---|---|---|---|
| UNCLASSIFIED | UNCLASSIFIED | UNCLASSIFIED | SAR |

## GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and its title page. Instructions for filling in each block of the form follow. It is important to ***stay within the lines*** to meet ***optical scanning requirements***.

**Block 1.** Agency Use Only *(Leave blank)*.

**Block 2.** Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

**Block 3.** Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

**Block 4.** Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

**Block 5.** Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

| | | | |
|---|---|---|---|
| C | - Contract | PR | - Project |
| G | - Grant | TA | - Task |
| PE | - Program Element | WU | - Work Unit Accession No. |

**BLOCK 6.** Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

**Block 7.** Performing Organization Name(s) and Address(es). Self-explanatory.

**Block 8.** Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

**Block 9.** Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

**Block 10.** Sponsoring/Monitoring Agency Report Number. *(If Known)*

**Block 11.** Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

**Block 12a.** Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

- **DOD** - See DoDD 5230.24, "Distribution Statements on Technical Documents."
- **DOE** - See authorities.
- **NASA** - See Handbook NHB 2200.2
- **NTIS** - Leave blank.

**Block 12b.** Distribution Code.

- **DOD** - Leave blank.
- **DOE** - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.
- **NASA** - Leave blank.
- **NTIS** - Leave blank.

**Block 13.** Abstract. Include a brief ***(Maximum 200 words)*** factual summary of the most significant information contained in the report.

**Block 14.** Subject Terms. Keywords or phrases identifying major subjects in the report.

**Block 15.** Number of Pages. Enter the total number of pages.

**Block 16.** Price Code. Enter appropriate price code ***(NTIS only)***

**Blocks 17.-19.** Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

**Block 20.** Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.