

IEEE Standard for Application and Management of the Systems Engineering Process

Sponsor

**Software Engineering Standards Committee
of the
IEEE Computer Society**

Approved 8 December 1998

IEEE-SA Standards Board

Abstract: The interdisciplinary tasks, which are required throughout a system's life cycle to transform customer needs, requirements, and constraints into a system solution, are defined. In addition, the requirements for the systems engineering process and its application throughout the product life cycle are specified. The focus of this standard is on engineering activities necessary to guide product development while ensuring that the product is properly designed to make it affordable to produce, own, operate, maintain, and eventually to dispose of, without undue risk to health or the environment.

Keywords: component, hardware, interface, life cycle processes, software, system life cycle, systems engineering

The Institute of Electrical and Electronics Engineers, Inc.
345 East 47th Street, New York, NY 10017-2394, USA

Copyright © 1999 by the Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 22 January 1999. Printed in the United States of America.

Print: ISBN 0-7381-1543-6 SH94720
PDF: ISBN 0-7381-1544-4 SS94720

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. Members of the committees serve voluntarily and without compensation. They are not necessarily members of the Institute. The standards developed within IEEE represent a consensus of the broad expertise on the subject within the Institute as well as those activities outside of IEEE that have expressed an interest in participating in the development of the standard.

Use of an IEEE Standard is wholly voluntary. The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE Standard is subjected to review at least every five years for revision or reaffirmation. When a document is more than five years old and has not been reaffirmed, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE Standard.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of all concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration.

Comments on standards and requests for interpretations should be addressed to:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
USA

Note: Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. The IEEE shall not be responsible for identifying patents for which a license may be required by an IEEE standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

Authorization to photocopy portions of any individual standard for internal or personal use is granted by the Institute of Electrical and Electronics Engineers, Inc., provided that the appropriate fee is paid to Copyright Clearance Center. To arrange for payment of licensing fee, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; (978) 750-8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Introduction

(This introduction is not part of IEEE Std 1220-1998, IEEE Standards for Application and Management of the Systems Engineering Process.)

History

This standard was initially published in January 1995 as a trial-use standard. During the two-year trial-use period, users of the standard gained experience with applying the document within their enterprises. In January 1998, the document was revised and balloted for full publication. The comments submitted by the balloters were considered, and those within the scope of the standard were incorporated. The revision was completed in July 1998 and submitted to the IEEE Standards Office for publication.

The key differences between this version of the standard and the trial-use version are

- a) Increased emphasis on engineering the system for humans (operator and maintainer);
- b) Partitioning functional analysis into context analysis and functional decomposition to provide a clearer depiction of the functional analysis process in support of requirements analysis and synthesis;
- c) Improved clauses to make conformance statements clearer;
- d) Simplified systems breakdown structure;
- e) Increased emphasis on software.

Purpose

This standard defines the requirements for an enterprise's total technical effort related to development of products (including computers and software) and processes that will provide life cycle support (sustain and evolve) for the products. It prescribes an integrated technical approach to engineering a system and requires the application and management of the systems engineering process throughout a product life cycle. The systems engineering process is applied recursively to the development or incremental improvement of a product to satisfy market requirements, and to simultaneously provide related life cycle processes for product development, manufacturing, test, distribution, operation, support, training, and disposal.

The concept of systems engineering embodied in this document provides an approach for product development in a system context. It is not meant to describe what an organizational entity called systems engineering does; or a job position for which a systems engineer is responsible. Rather, it encompasses what all organizational entities and all enterprise personnel must accomplish to produce a quality, competitive product that will be marketable, will provide an acceptable return on investment to the enterprise, will achieve customer satisfaction, and will meet public expectations.

The fundamental systems engineering objective is to provide high-quality products and services, with the correct people and performance features, at an affordable price, and on time. This involves developing, producing, testing, and supporting an integrated set of products (hardware, software, people, data, facilities, and material) and processes (services and techniques) that is acceptable to customers, satisfies enterprise and external constraints, and considers and defines the processes for developing, producing, testing, handling, operating, and supporting the products and life cycle processes. This objective is achieved by simultaneous treatment of product and process content to focus project resources and design decisions for the establishment of an effective system design. This involves an integrated handling of all elements of a system, including those related to manufacturing, test, distribution, operations, support, training, and disposal.

Participants

At the time this standard was completed, the Working Group on Systems Engineering Management had the following membership:

Richard F. Schmidt , <i>Chair</i>	Kenneth R. Ptack , <i>Editor</i>	
Jerry Lake , <i>NCOSE Liaison</i>	Jim Lacy , <i>EIA Liaison</i>	
George Friedman , <i>AIS Liaison</i>		
Jim Armstrong	Bill Gess	John Snodderly
	Joel O'Rourke	

Other individuals who have contributed to the development of this standard are

Stuart Arnold	Jennifer McKneely	Arnold M. Ruskin
Randy Case	Amie Perry	Jack Sivak
Jim Cloud	Peter T. Poon	Ronald L. Wade
Roger D. Larson		John Winters

Other individuals who have contributed reviews and comments for this standard are

Frank Akerman	Michael P. Guba	Alfred H. Peschel
Edward E. Barnett	David A. Gustafson	Robert M. Poston
H. Ronald Berlack	Jon D. Hagar	H. Reiche
Richard E. Biehl	Ramzi Haraty	Gregory Russel
Mark Bilger	Manfred Hein	Ronmald K. Saltgaver
William J. Boll, Jr.	John W. Horch	Julio G. Sanz
Ronald Braun	David Johnson	David J. Schultz
Bruce Brocka	Frank V. Jorgensen	Gregory D. Schumacher
Fletcher Buckley	Diana Kang	Leonard W. Seagren
Susan M. Burgess	Richard Karcich	Carl S. Seddio
Jaya R. Carl	Ron S. Kenett	J. J. Selis
Niva M. Carlson	Judith S. Kerner	Keith Shewbridge
Basilio Chen	Scotty Kilbourne	Robert W. Shillato
Hu Cheng	F. L. King	David M. Siefert
John Chihonek	Dwayne L. Knirk	Mitchell W. Smith
Tsun S. Chow	Ed Kulas	Harry M. Sneed
Lee Cooper	John B. Lane	Alfred R. Sorkowitz
Taz Daughtrey	Boniface Lau	Vijaya K. Srivastava
Paul I. Davis	J. Dennis Lawrence	Richard Staunton
Bostjan K. Derganc	Michael Lines	Jim Swanek
Einar Dragstedt	William M. Lively	Douglas H. Thiele
Scott P. Duncan	Ben Livson	Leonard L. Tripp
Peter Eirich	James J. Longbucco	Margaret Updike
Mokhtar S. El Assad	Joseph Maayan	Ted J. Urbanowicz
Richard Evans	Jukka Marijarvi	Glenn D. Venables
John W. Fendrich	James W. McClean	Peter Voldner
Jay Forster	Patrick D. McCray	L. Mark Walker
Richard C. Fries	Sue McGrath	Delores Wallace
Simon Gabrihelidis	Glen A. Meldrum	John W. Walz
Yair Gershkovitch	James E. Miller	Camille White-Partain
Gregg Giesler	Louis E. Miller	Paul R. Work
April Gillam	Celia H. Modell	Forrest D. Wright
Shirley A. Gloss-Soler	Hironobu Nagano	Natalie C. Yopconka
John Garth Glynn	Dennis E. Nickle	Wieder D. Yu
Rajeev Goel	Mike Ottewill	Janusz Zalewski
Donald Gotterbarn	Robert Parys	Peter F. Zoll
	David E. Peercy	

In addition, this standard was coordinated with the following organizations:

AIA
ASC X3
ASQ

Department of Defense
EIA
ISO/IEC/JTC1/SC7

MSC
NNCC
SSC10

The following members of the balloting committee voted on this standard:

Leo Beltracchi
H. Ronald Berlack
Richard E. Biehl
Juris Borzovs
Antonio M. Cicu
Francois Coallier
Virgil Lee Cooper
Thomas Crowley
Geoffrey Darnton
Taz Daughtrey
Raymond Day
Bostjan K. Derganc
Perry R. DeWeese
Charles Droz
Sherman Eagles
Leo G. Egan
William Eventoff
Jonathan H. Fairclough
Richard E. Fairley
John W. Fendrich
Jay Forster
Kirby Fortenberry
Karol Fruehauf
Marilyn Ginsberg-Finner
John Garth Glynn
Julio Gonzalez-Sanz
L. M. Gunther
David A. Gustafson
Jon D. Hagar
John Harauz
Rob Harker

Herbert Hecht
William Hefley
Debra Herrmann
John W. Horch
Jerry Huller
George Jackelen
Frank V. Jorgensen
Vladan V. Jovanovic
William S. Junk
Ron S. Kenett
Judith S. Kerner
Robert J. Kierzyk
Thomas M. Kurihara
John B. Lane
J. Dennis Lawrence
Randal Leavitt
William M. Lively
Dieter Look
Stan Magee
Harold Mains
Robert A. Martin
Patrick D. McCray
Jerome W. Mersky
Alan Miller
James W. Moore
Pavol Navrat
Donald J. Ostrom
Mike Ottewill
John G. Phippen
Alex Polack

Peter T. Poon
Kenneth R. Ptack
Donald J. Reifer
R. Waldo Roth
Andrew P. Sage
Helmut Sandmayr
Stephen R. Schach
Norman Schneidewind
David J. Schultz
Robert W. Shillato
Carl A. Singer
Mitchell W. Smith
Melford E. Smyre
Alfred R. Sorkowitz
Julia Stesney
Fred J. Strauss
Toru Takeshita
Richard H. Thayer
Booker Thomas
Patricia Trelue
Leonard L. Tripp
Theodore J. Urbanowicz
Glenn D. Venables
David D. Walden
Delores Wallace
Scott A. Whitmire
Paul A. Wolfgang
Paul R. Work
Forrest D. Wright
Janusz Zalewski
Geraldine Zimmerman

When the IEEE-SA Standards Board approved this standard on 8 December 1998, it had the following membership:

Richard J. Holleman, *Chair*

Donald N. Heirman, *Vice Chair*

Judith Gorman, *Secretary*

Satish K. Aggarwal
Clyde R. Camp
James T. Carlo
Gary R. Engmann
Harold E. Epstein
Jay Forster*
Thomas F. Garrity
Ruben D. Garzon

James H. Gurney
Jim D. Isaak
Lowell G. Johnson
Robert Kennelly
E. G. "Al" Kiener
Joseph L. Koepfinger*
Stephen R. Lambert
Jim Logothetis
Donald C. Loughry

L. Bruce McClung
Louis-François Pau
Ronald C. Petersen
Gerald H. Peterson
John B. Posey
Gary S. Robinson
Hans E. Weinrich
Donald W. Zipse

*Member Emeritus

Yvette Ho Sang
IEEE Standards Project Editor

Contents

1.	Overview	1
1.1	Scope	1
1.2	Purpose	2
1.3	Conformance	2
1.4	How to use this standard	2
1.5	Organization of this standard	5
1.6	Assumptions of this standard	5
2.	References	5
3.	Definitions and acronyms	5
3.1	Definitions	5
3.2	Acronyms	9
4.	General requirements	9
4.1	Systems engineering process (SEP)	10
4.2	Policies and procedures for systems engineering	10
4.3	Planning the technical effort	10
4.4	Development strategies	12
4.5	Modeling and prototyping	12
4.6	Integrated database	12
4.7	Integrated data package	12
4.8	Specification tree	14
4.9	Drawing tree	14
4.10	System breakdown structure (SBS)	14
4.11	Integration of the systems engineering effort	15
4.12	Technical reviews	16
4.13	Quality management	16
4.14	Product and process improvement	16
5.	Application of systems engineering throughout the system life cycle	17
5.1	System definition stage	18
5.2	Preliminary design stage	22
5.3	Detailed design stage	26
5.4	Fabrication, assembly, integration, and test (FAIT) stage	30
5.5	Production and customer support stages	32
5.6	Simultaneous engineering of life cycle processes	33
6.	The systems engineering process (SEP)	34
6.1	Requirements analysis	34
6.2	Requirements validation	40
6.3	Functional analysis	42
6.4	Functional verification	45
6.5	Synthesis	46
6.6	Design verification	50
6.7	Systems analysis	54
6.8	Control	58

Annex A (informative) The role of systems engineering within an enterprise	64
Annex B (informative) The systems engineering management plan	68
Annex C (informative) Bibliography	76

IEEE Standard for Application and Management of the Systems Engineering Process

1. Overview

1.1 Scope

This standard defines the interdisciplinary tasks that are required throughout a system's life cycle to transform customer needs, requirements, and constraints into a system solution. This document is intended to guide the development of systems (which include humans, computers, and software) for commercial, government, military, and space applications. The information applies to an enterprise within an enterprise that is responsible for developing a product design and establishing the life cycle infrastructure needed to provide for life cycle sustainment.

This standard specifies the requirements for the systems engineering process (SEP) and its application throughout the product life cycle. It does not attempt to define the implementation of each system life cycle process, but addresses the issues associated with defining and establishing supportive life cycle processes early and continuously throughout product development. In addition, the standard does not address the many cultural or quality variables, which should be considered for successful product development. The standard focuses on the engineering activities necessary to guide product development while ensuring that the product is properly designed to make it affordable to produce, own, operate, maintain, and eventually to dispose of, without undue risk to health or the environment.

The requirements of this standard are applicable to new products as well as incremental enhancements to existing products. It applies to one-of-a-kind products, such as a satellite, as well as products that are mass produced for the consumer marketplace. The requirements of this standard should be selectively applied for each specific system-development project. The role of systems engineering within the enterprise environment is described in Annex A.

The content of this standard describes an integrated approach to product development, which represents the total technical effort for

- a) Understanding the environments and the related conditions in which the product may be utilized, and for which the product should be designed to accommodate;
- b) Defining product requirements in terms of functional and performance requirements, quality factors, usability, producibility, supportability, safety, and environmental impacts;
- c) Defining the life cycle processes for manufacturing, test, distribution, support, training, and disposal, which are necessary to provide life cycle support for products.

1.2 Purpose

The purpose of this document is to provide a standard for managing a system from initial concept through development, operations, and disposal. The inclusion of computers and associated software in today's products has made the need to engineer each of those products as a total system more acute. The human, physical, and software components should all be addressed to optimize overall system performance.

1.3 Conformance

Conformance with this standard is defined as the establishment of enterprise procedures, and their execution, for the SEP (Clause 6) and its application throughout the system life cycle (Clause 5). Individual clauses are mandatory when the clause includes a "shall" statement. Clauses are recommended when the clause includes a "should" statement. Clauses are optional when the clause includes a "may" statement. To be compliant, an enterprise should establish procedures for accomplishing all mandatory clauses, and selected recommended and optional clauses, and should ensure that each project within the enterprise complies with these procedures. Clause 4 provides some generic requirements for implementing systems engineering within an enterprise or on a project. These requirements invoke a mandatory "shall" when the clause is mandatory.

Clause 6 defines the SEP, which is accomplished iteratively to define system products and life cycle processes. Thus, the initial clause defining the subprocesses (requirements analysis, functional analysis, etc.) includes a mandatory "shall" statement to ensure that an enterprise's SEP addresses each of the subprocesses. The remaining subclauses, when specified within the definition of the subprocesses, include a recommended "should" statement to provide flexibility in adapting the subprocesses to the purpose of the current iteration of the SEP. During any particular iteration of the SEP, not all of the subclauses need to be accomplished to be in conformance with this standard. Therefore, conformance with Clause 6 requires an enterprise to establish policies and procedures for accomplishing the subclauses within the subprocesses.

1.4 How to use this standard

The description of the SEP, and its application throughout the life cycle, demands the use of a system paradigm to aid the presentation of this material. The terms used to support this paradigm are defined in Clause 3. As enterprises gain familiarity with the paradigm, they may substitute more familiar terms that are applicable to their industry or business practices. The system paradigm is the foundation of this standard and is described below to support the different uses of the term *system*.

On a large scale, there are biological systems, ecological systems, weather systems, solar systems, etc. Thus, a system can be viewed as an element of a larger system, and the challenge is to understand the boundary of the system, which is the focus of the development effort, and the relationships and interfaces between this system and other systems. The focus of this standard is product-oriented systems such as the automobile, the airplane, or information systems.

A system is typically composed of related elements (subsystems and components) and their interfaces. Additionally, elements include the people required to develop, produce, test, distribute, operate, support, or dispose of the element's products, or to train those people to accomplish their role within the system. Figure 1 provides a hierarchy of names for the elements making up a system. This generic system hierarchy is a key concept within this standard because it ties the system architectures, specification and drawing trees, system breakdown structure (SBS), technical reviews, and configuration baselines together. Many elements within the system hierarchy can be considered a "system" by the classical definition, but actually represent subsystems within the system hierarchy. Likewise, the life cycle processes represent subsystems within the overall system hierarchy.

Complex components represent system elements that are composed of hardware, software, and/or humans, which are recognizable in terms of life cycle process (how to design, test, produce, support, etc., is known), and the domain-specific engineering team assumes responsibility for the development of the complex

components. This is a judgement call—complex components may demand the rigors of the SEP or may be well suited for component development teams.

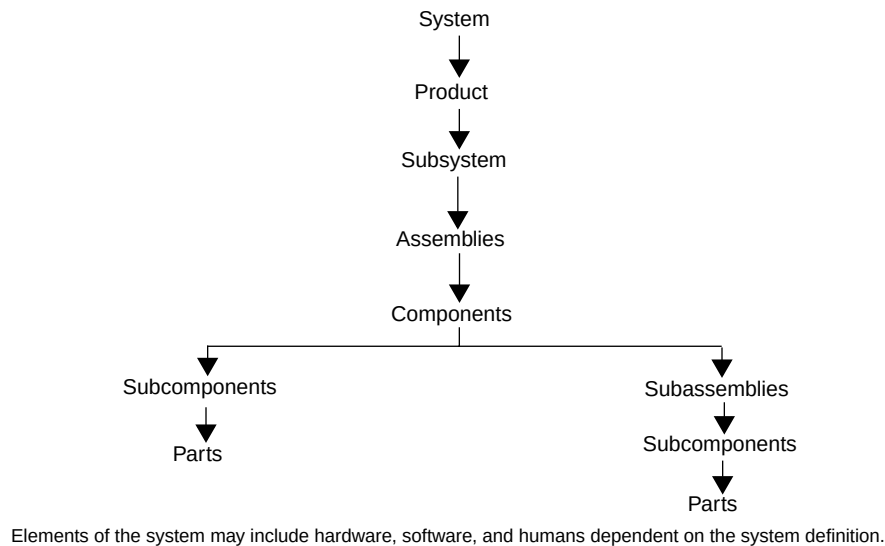


Figure 1—Hierarchy of elements within a system

The human elements are integral to the systems hierarchy, and may be present at any level. The human elements are not identified in the system hierarchy since the intent of the hierarchy is to identify the system element for which the system is being defined, and the human/system integration issues should be addressed in terms of the human's role in operating, producing, supporting, etc., the system.

The hierarchy of elements within a system is provided to illustrate that systems may be comprised of other systems (subsystems), which represent complex elements for which no existing design solution or supplier can be identified. The number of levels of subsystem or complex components is dependent on the complexity of the system being developed. The SEP is applied at each level in the system hierarchy for which the system element is a complex item for which no available design solution, or existing producer, can be identified. Once a system element can be identified with a hardware, software, or human element, the discipline-specific design methodologies are utilized to design the system element.

The basic building blocks of a system are depicted in Figure 2—the system, its related product(s), the life cycle processes required to support the products, and the subsystems that make up the product(s). Each life cycle process—development, manufacturing, test, distribution, operation, support, training, or disposal—is itself like a system in that products should be developed to fulfill the purpose of the life cycle process. For example, a product should be manufactured. Manufacturing is a life cycle process. The products associated with the manufacturing life cycle process include special equipment, tools, facilities, and production processes and procedures. The products that make up life cycle processes may also require life cycle sustainment in that they may be required to be developed, tested, manufactured, distributed, operated, supported, trained, and disposed of.

A typical system is composed of products developed by the enterprise or by suppliers/subcontractors. Each supplier/subcontractor considers its product as part of its system. The organization that purchases these products for integration into a higher-level system should refer to these products as subcomponents, components, complex components, or subsystems, depending on the significance of the element in contributing to the system's performance, functionality, and costs.

The design of the products and life cycle processes should consider the human as an element of the system in terms of operators, maintainers, manufacturing personnel, training personnel, etc., for the purpose of understanding the human/system integration issues and ensuring that the system products are producible,

maintainable, and usable, and that the system processes are effectively established to ensure production quality levels and reduce overall ownership cost. Thus, Figure 3 depicts the human elements associated with the system products and processes. The operational process addresses the operation of the product(s) and aids in the identification of operational procedures and human cognitive and anthropomorphic considerations necessary to ensure system usability.

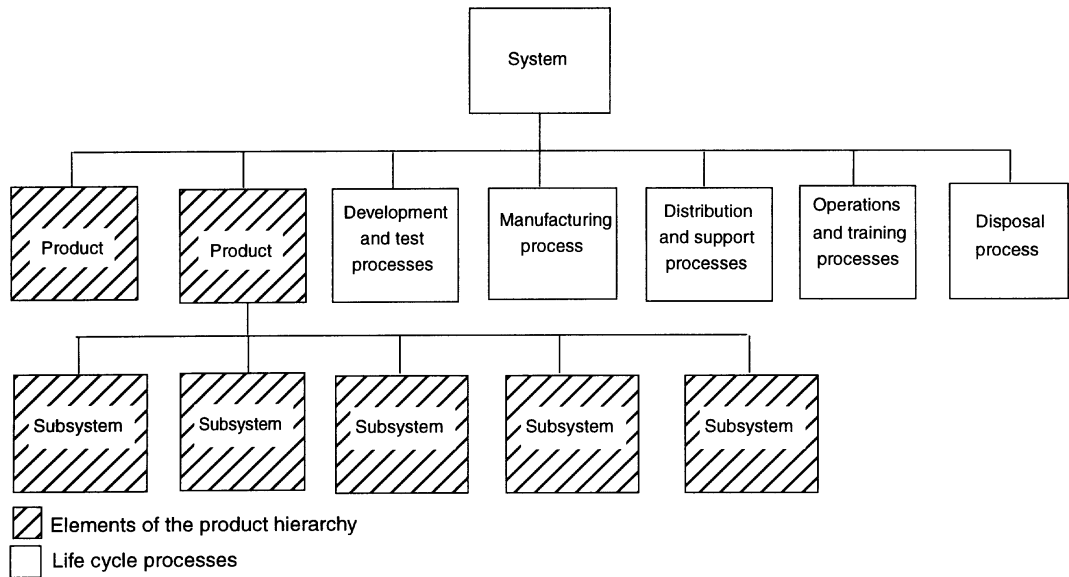


Figure 2—Basic building blocks of a system

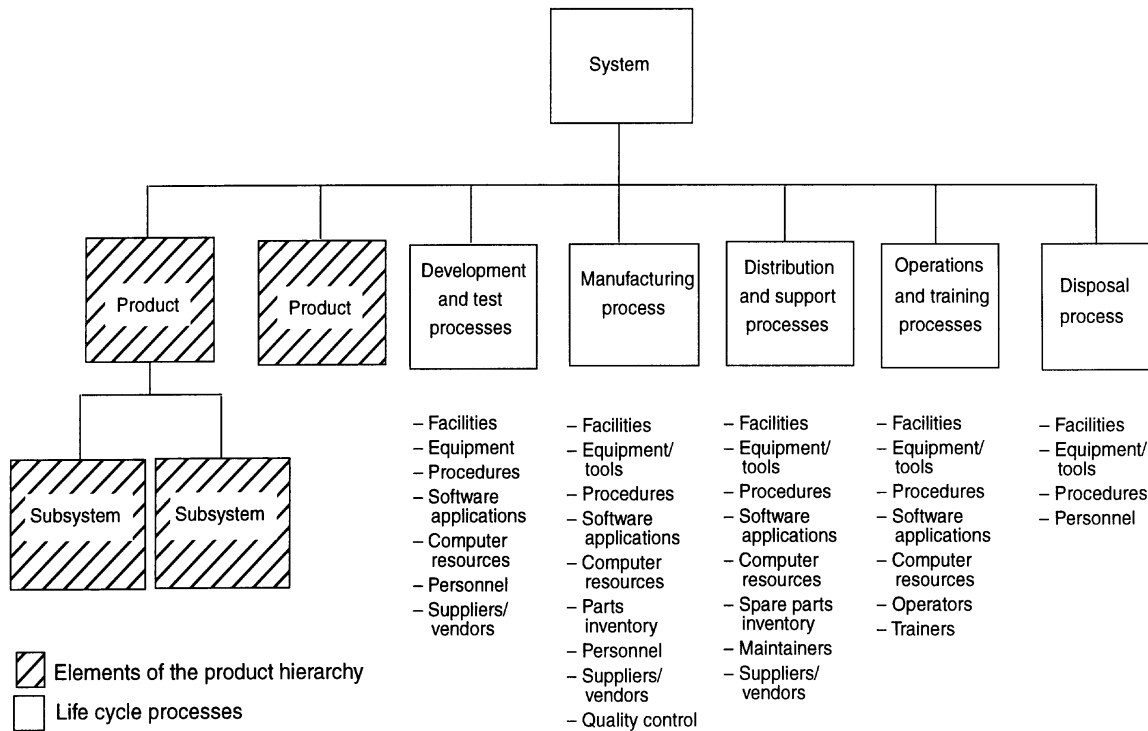


Figure 3—Life cycle process definition

The definitions of system elements are generated by activities of the SEP. This process is described in detail in Clause 6. The SEP is used during each level of development to structure systems engineering activities that identify technical requirements and desired system behaviors, and synthesize the system design.

1.5 Organization of this standard

- Clause 1 provides the scope, purpose, and organization of this standard.
- Clause 2 provides the references applicable to this document.
- Clause 3 establishes the meaning of terms and acronyms, as used in this standard.
- Clause 4 establishes requirements for planning and implementing an effective systems engineering capability within an enterprise.
- Clause 5 provides a description of the application of the SEP through system definition, subsystem definition, production, and customer support.
- Clause 6 provides the detailed tasks of the SEP to be tailored and performed to develop product solutions and their supporting life cycle processes.
- Annex A discusses the SEP as the total technical effort responsible for establishing the product design and life cycle support products within an enterprise.
- Annex B provides a template to help an enterprise prepare a systems engineering management plan.
- Annex C provides bibliographical references.

1.6 Assumptions of this standard

This standard assumes that the user has a quality management system with capability that meets the requirements of ISO 9001: 1994 [B2].¹ This standard also assumes that the organization responsible for the software as the result of the use of this standard shall have implemented a set of processes that meet the requirements of IEEE/EIA 12207.0-1996 [B1].

2. References

This standard shall be used in conjunction with the following publications. If the following publications are superseded by an approved revision, the revision shall apply.

IEEE Std 100-1996, IEEE Standard Dictionary of Electrical and Electronics Terms.²

IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology.

3. Definitions and acronyms

3.1 Definitions

The definitions listed below establish meaning in the context of this standard. If there is a conflict between the terms in the references and this standard, then the definitions provided here should take precedence as they apply only to the practices and requirements of this standard.

¹The numbers in brackets correspond to those of the bibliography in Annex C.

²IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, USA (<http://www.standards.ieee.org/>).

3.1.1 allocation: The decision to assign a function or decision to hardware, software, or humans. Allocation may be made entirely to one of these three system element types or to some combination to be resolved upon further functional decomposition.

3.1.2 behavioral analysis: The analysis of the logical (stimulus/response) and design (resource consumption, event timing, throughput, etc.) execution of a system to assess the functional and design architectures.

3.1.3 concurrent engineering: The simultaneous engineering of products and life cycle processes to ensure usability, producibility, and supportability, and to control life cycle and total ownership costs.

3.1.4 configuration baseline: The configuration at a point in time recorded in documentation that fully describes the functional, performance, usability, interface requirement, and design characteristics, as appropriate to the stage of the life cycle.

3.1.5 constraint: A limitation or implied requirement that constrains the design solution or implementation of the systems engineering process, is not changeable by the enterprise, and is generally nonallocable.

3.1.6 customer: A person or organization ordering, purchasing, receiving, or affected by a product or process provided by the enterprise. Customers include developers, manufacturers, testers, distributors, operators, supporters, trainers, disposers, and the general public.

3.1.7 design architecture: An arrangement of design elements that provides the design solution for a product or life cycle process intended to satisfy the functional architecture and the requirements baseline.

3.1.8 design characteristic: The design attributes or distinguishing features that pertain to a measurable description of a product or process.

3.1.9 design to cost: Cost goals for the production cost (design to unit production cost) and life cycle cost (design to life cycle cost) of the products used to make the design converge on cost targets, and is equivalent to any other performance parameter.

3.1.10 design verification: The process of evaluating whether or not the requirements of the design architecture are traceable to the verified functional architecture and satisfy the validated requirement baseline.

3.1.11 effectiveness analysis: An analysis of how well a design solution will perform or operate given anticipated operational scenarios.

3.1.12 effectiveness assessment: The evaluation of the design solution with respect to manufacturing, test, distribution, operations, support, training, environmental impact, cost effectiveness, and life cycle cost.

3.1.13 effectiveness criteria: The measure of value used to determine the success or failure of a design solution.

3.1.14 end item: An entity identified with an element of the system breakdown structure. An end item is represented by one or more of the following: equipment (hardware and software), data, facilities, material, services, and/or techniques.

3.1.15 enterprise: The organization that performs specified tasks.

3.1.16 event-based planning: An approach to establishing engineering plans, tasks, and milestones based upon satisfying significant accomplishments associated with key events rather than calendar-oriented milestones.

3.1.17 functional architecture: An arrangement of functions and their subfunctions and interfaces (internal and external) that defines the execution sequencing, conditions for control or data flow, and the performance requirements to satisfy the requirements baseline.

3.1.18 functional requirement: A statement that identifies what a product or process must accomplish to produce required behavior and/or results.

3.1.19 functional verification: The process of evaluating whether or not the functional architecture satisfies the validated requirements baseline.

3.1.20 human systems engineering: The activities involved throughout the system life cycle that addresses the human element of system design (including usability, measures of effectiveness, measures of performance, and total ownership cost). These activities include the definition and synthesis of manpower, personnel, training, human engineering, health hazards, and safety issues.

3.1.21 integrated database: A repository for storing all information pertinent to the systems engineering process to include all data, schema, models, tools, technical management decisions, process analysis information, requirement changes, process and product metrics, and trade-offs.

3.1.22 integrated data package: The evolving output of the systems engineering process that documents hardware, software, life cycle processes, and human engineering designs.

3.1.23 interface specification: The description of essential functional, performance, and design requirements and constraints at a common boundary between two or more system elements. This includes interfaces between humans and hardware or software, as well as interfaces between humans themselves.

3.1.24 life cycle: The system or product evolution initiated by a perceived customer need through the disposal of the products.

3.1.25 life cycle cost: The total investment in product development, manufacturing, test, distribution, operation, support, training, and disposal.

3.1.26 life cycle processes: The following eight essential functional processes that may be necessary to provide total consumer satisfaction and meet public acceptance. Once the need for a life cycle process is identified, the life cycle process is treated as a system, and the systems engineering process is applied to define, design, and establish the life cycle process, and the supporting products and processes to maintain the life cycle process in an operational condition.

- a) *Development.* The planning and execution of system and subsystem definition tasks required to evolve the system from customer needs to product solutions and their life cycle processes.
- b) *Manufacturing.* The tasks, actions, and activities for fabrication and assembly of engineering test models and brassboards, prototypes, and production of product solutions and their life cycle process products.
- c) *Test.*
 - 1) The tasks, actions, and activities for planning for evaluation and conducting evaluation of synthesis products against the functional architecture or requirements baseline, or the functional architecture against the requirements baseline.
 - 2) The tasks, actions, and activities for evaluating the product solutions and their life cycle processes to measure specification conformance or customer satisfaction.
- d) *Distribution.* The tasks, actions, and activities to initially transport, deliver, assemble, install, test, and checkout products to effect proper transition to users, operators, or consumers.
- e) *Operations.* The tasks, actions, and activities that are associated with the use of the product or a life cycle process.

- f) *Support*. The tasks, actions, and activities to provide supply, maintenance, and support material and facility management for sustaining operations.
- g) *Training*. The measurable tasks, actions, and activities (including instruction and applied exercises) required to achieve and maintain the knowledge, skills, and abilities necessary to efficiently and effectively perform operations, support, and disposal throughout the system life cycle. Training is inclusive of the tools, devices, techniques, procedures, and materials developed and employed to provide training for all required tasks.
- h) *Disposal*. The tasks, actions, and activities to ensure that disposal or recycling of destroyed or irreparable consumer and life cycle processes and by-products comply with applicable environmental regulations and directives.

3.1.27 measure of effectiveness (MOE): The metrics by which a customer will measure satisfaction with products produced by the technical effort.

3.1.28 measure of performance (MOP): An engineering performance measure that provides design requirements that are necessary to satisfy a measure of effectiveness. There are generally several measures of performance for each measure of effectiveness.

3.1.29 performance requirement: The measurable criteria that identifies a quality attribute of a function, or how well a functional requirement must be accomplished.

3.1.30 precedented system: A system for which design examples exist within its class, so as to provide guidance for establishing the design architecture, engineering and technical plans, specifications, or low risk alternatives.

3.1.31 project environment: An environment that defines the objectives, success criteria, project milestones, and associated management priorities that govern the systems engineering activities in support of product development.

3.1.32 re-engineering: The process of improving a system after production through modification to correct a design deficiency or to make an incremental improvement.

3.1.33 requirement: A statement that identifies a product or process operational, functional, or design characteristic or constraint, which is unambiguous, testable or measurable, and necessary for product or process acceptability (by consumers or internal quality assurance guidelines).

3.1.34 specification: A document that fully describes a design element or its interfaces in terms of requirements (functional, performance, constraints, and design characteristics) and the qualification conditions and procedures for each requirement.

3.1.35 specification tree: A hierarchy of specification elements and their interface specifications that identifies the elements and the specifications related to design elements of the system configuration which are to be controlled.

3.1.36 state: A condition that characterizes the behavior of a function/subfunction or element at a point in time.

3.1.37 system: A set or arrangement of elements [people, products (hardware and software) and processes (facilities, equipment, material, and procedures)] that are related and whose behavior satisfies customer/operational needs, and provides for the life cycle sustainment of the products.

3.1.38 system architecture: The composite of the design architectures for products and their life cycle processes.

3.1.39 system breakdown structure (SBS): A hierarchy of elements, related life cycle processes, and personnel used to assign development teams, conduct technical reviews, and to partition out the assigned work and associated resource allocations to each of the tasks necessary to accomplish the objectives of the project. It also provides the basis for cost tracking and control.

3.1.40 system effectiveness: A measurement of the ability of a system to satisfy its intended operational uses as a function of how the system performs under anticipated environmental conditions, and the ability to produce, test, distribute, operate, support, train, and dispose of the system throughout its life cycle.

3.1.41 trade-off analysis: An analytical evaluation of design options/alternatives against performance, design-to-cost objectives, and life cycle quality factors.

3.1.42 unprecedented system: A system for which design examples do not exist so that the design architecture alternatives are unconstrained by previous system descriptions.

3.2 Acronyms

Term definitions are consistent with IEEE Std 610.12-1990.³

BIT	Built-in test
FAIT	Fabrication, assembly, integration, and test
FIT	Fault-isolation test
FMEA	Failure, modes, and effects analysis
IPT	Integrated product team
MOE	Measure of effectiveness
MOP	Measure of performance
SBS	System breakdown structure
SEP	Systems engineering process
TPM	Technical performance measure

4. General requirements

The enterprise shall plan, implement, and control an integrated technical effort in accordance with this document to develop a total system solution that is responsive to market opportunities, specified customer requirements, enterprise objectives, and external constraints. To meet this goal, the enterprise

- a) Shall plan, conduct, and manage a fully integrated technical effort necessary to satisfy the general requirements of this document, as tailored for the specific project;
- b) Shall apply the SEP for each level of system decomposition (e.g., concept definition; system definition; subsystem design; fabrication, assembly, integration, and test; production; and sustainment);
- c) Shall control progress through the conduct of
 - 1) Technical reviews following each level of development;
 - 2) Risk management;
 - 3) Data management;
 - 4) Interface management;
 - 5) Configuration management;
 - 6) Performance-based progress measurement;

³Information on references can be found in Clause 2.

- d) Shall generate models and prototypes to support trade-off;
- e) Shall generate an integrated data package, which ensures that the product can be produced, tested, delivered, operated, supported, and properly disposed of;
- f) Should capture the outputs from all technical activities in an integrated database.

4.1 Systems engineering process (SEP)

The enterprise shall apply the SEP described in Clause 6, and illustrated in Figure 4, to produce the specifications and baselines and related products, as described in Clause 5. The SEP is a generic problem-solving process, which provides the mechanisms for identifying and evolving the product and process definitions of a system. The SEP applies throughout the system life cycle to all activities associated with product development, verification/test, manufacturing, training, operation, support, distribution, disposal, and human systems engineering. Figure 4 depicts the elements of the SEP and shows how they iterate to produce a consistent set of requirements, functional arrangements, and design solutions. The activities of this process are described in detail in Clause 6.

4.2 Policies and procedures for systems engineering

The enterprise shall develop and maintain policies and procedures for governing the conduct of the SEP. These policies and procedures specify requirements for the planning, implementation, and control of product and process development and human/systems integration. These policies and procedures should establish the enterprise's paradigm for systems engineering from which training can be based and project specific activities can be accomplished. Enterprise policies and procedures shall address the following:

- a) The application of the SEP throughout a project life cycle;
- b) The preparation and approval of a systems engineering management plan;
- c) The preparation and approval of a systems engineering master schedule (SEMS) and systems engineering detailed schedule;
- d) The preparation and approval of the integrated data package;
- e) Monitoring and reporting technical progress of the project;
- f) The preparation for, and conduct of, technical reviews;
- g) The contents and maintenance of an integrated database;
- h) The continuous improvement of products and processes.

4.3 Planning the technical effort

The enterprise shall prepare and implement the technical plans and schedules necessary to guide the project toward accomplishment of its objectives and proper conclusion. Given the project's authorization and objectives, the enterprise should establish an engineering plan, a master schedule, and a detail schedule. The engineering plan should be the main planning document for all technical efforts and describes the tailored application of this standard. The master schedule, an event-based schedule, and the detail schedule, a calendar-based schedule derived from the master schedule, should address the development activities for the product, as well as supporting life cycle processes. The master schedule and detail schedule may be combined into a single engineering schedule. The enterprise should establish plans for the technical effort in accordance with the following subclauses.

4.3.1 Engineering plan

The engineering plan shall be prepared and updated throughout the system life cycle to guide and control the technical efforts of the project. The engineering plan should reflect an integrated technical effort responsible for product development, which addresses all factors associated with meeting system life cycle requirements.

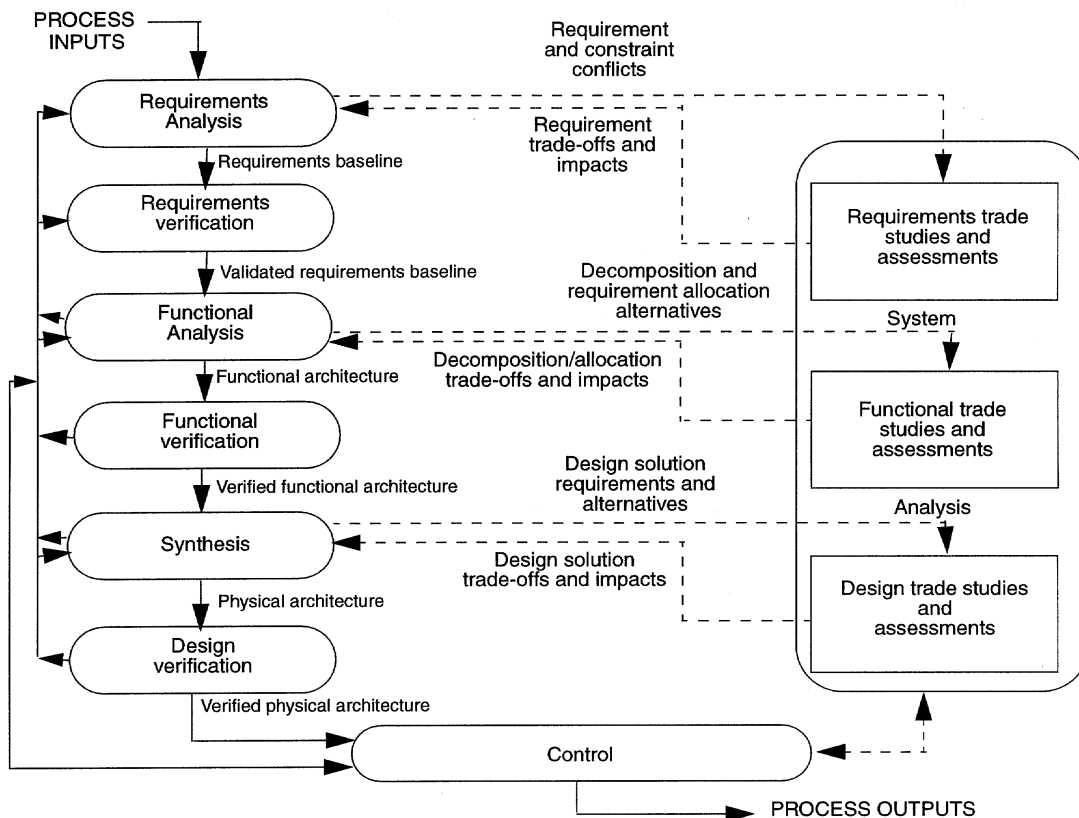


Figure 4—Systems engineering process (SEP)

The engineering plan should reflect the contents identified in Annex B, which provides a generic outline and description of the contents for an engineering plan. If an evolutionary or incremental development strategy is to be pursued, the engineering plan should address the development strategy for initial product development and insertion of incremental technology enhancements.

4.3.2 Master schedule

The master schedule should be prepared and updated throughout the system life cycle to establish key events, their significant tasks, and the criteria by which completion of significant tasks are determined. A properly designed master schedule should allow for the scheduling of activities associated with tasks, allocation of resources, budget preparation, assignment of personnel, establishment of task start and end dates, and event completion criteria.

4.3.3 Detail schedule

The detail schedule should provide a calendar-based schedule of activities, tasks, and key events of the master schedule. The detail schedule should be used to track the progress of the technical effort. Detail schedule data should be used to construct a network of events, tasks, and activities to determine the critical path of the engineering efforts and to analyze variances to schedule.

4.3.4 Technical plans

Technical plans shall be prepared to supplement the engineering plan, as needed. Technical plans should be prepared by the engineering and technical specialty areas to which they apply and should be used to measure technical progress against the plans. Technical plans are typically prepared for risk management, configura-

tion management, technical reviews, verifications, computer resources, manufacturing, maintenance, training, security, and human systems engineering.

4.4 Development strategies

The enterprise should explore development strategies for developing the system and its capabilities (e.g., waterfall, incremental, evolutionary, and spiral). The capacity to change or enhance the product and life cycle processes shall be designed into the system architecture to enable the cost-effective sustainment of the system throughout its life cycle. This design attribute shall be established early in the system development [when identifying customer or enterprise requirements or constraints (see 6.1.1 and 6.1.2)] to provide a basis for planning each incremental development effort. Evolutionary development strategies should address approaches for managing the introduction of new technologies, evolving requirements, or enhancing product capabilities.

4.5 Modeling and prototyping

The enterprise shall establish the models, simulations, or prototypes needed to support requirements definition, analyze the system architecture and design, mitigate identified risks, and thereby ensure that the final product satisfies market needs, requirements, and constraints. This effort supports the assessment of system functional and performance characteristics, producibility, supportability, environmental impact, and human systems engineering issues such as maintainability, usability, operability, and safety.

4.6 Integrated database

The enterprise shall capture design data in an integrated database to have a repository for the evolving integrated data package, and to provide a shared data source for the exchange and reuse of technical information.

4.6.1 Data and schema

The enterprise shall capture pertinent design data within an integrated database. These data include the engineering plan; schedules; technical plans; functional, physical, and system architectures; rationale for design decisions; trade-off analyses accomplished, including recommendations and impacts; effectiveness assessments and their outcomes; risk assessments and handling options; sketches and drawings; engineering changes; specification trees; specifications and configuration baselines; operational environment; manpower, personnel, training, and human engineering requirements and specifications; archival data; technical objectives, requirements, constraints, interfaces, and risks; SBSs; design models; test results; metrics; and any other data that allow for traceability of requirements throughout the functional and design architectures.

4.6.2 Tools

The enterprise shall maintain traceability between the versions of tools and the models, analysis results, and other outputs of tools to ensure the ability to reproduce execution results supporting the SEP.

4.7 Integrated data package

The enterprise shall generate an integrated data package, which documents architecture and design information for product manufacturing and life cycle sustainment, as required. The integrated data package should include the types of engineering data defined in Table 1, as required by the project. The specific content of the integrated data package shall be defined early in the system life cycle, should consist of the classes of information identified in 4.7.1 through 4.7.4, and should evolve with each application of the SEP to provide more detail with each level of development.

Table 1—Integrated data package content

An integrated data package includes, but is not limited to the following items:	
A. Hardware	
1)	Arrangement drawings —documents the relationship of the major subsystems or components of the system.
2)	Assembly drawings —documents the relationship of a combination of parts and subassemblies required to form the next higher indenture level of equipment or system.
3)	Connection drawings —documents the electrical connections of an installation or of its component devices or parts.
4)	Construction drawings —documents the design of buildings or structures.
5)	Product drawings —an engineering drawing that documents configuration and configuration limitations, performance and test requirements, weight and space requirements, access clearances, pipe and cable attachments, support requirements, etc., to the extent necessary that an item may be developed or procured on the commercial market to meet the stated requirements.
6)	Detail drawings —documents complete end-item requirements for the subcomponent(s) delineated in the drawing.
7)	Elevation drawings —documents vertical projections of buildings and structures or profiles of equipment.
8)	Engineering drawings —an engineering document that discloses, by means of pictorial or textual presentations, or a combination of both, the design and functional end-product requirements or design of an item.
9)	Installation drawings —documents general configuration and complete information necessary to install an item relative to its supporting structure or to associated items.
10)	Logic diagrams —documents, by means of graphic symbols or notations, the sequence and functions of logic circuitry and flows of sequences for operations, maintenance, test, and repair.
11)	Numerical control drawings —documents complete design and functional engineering and product requirements of an item to facilitate production by tape control means.
12)	Piping diagrams —documents the interconnection of components by piping, tubing, or hose; and when desired, the sequential flow of hydraulic fluids or pneumatic air in the system.
13)	Wire lists —documents a book-form drawing consisting of tabular data and instructions required to establish wiring connections within or between items.
14)	Schematic diagrams —documents, by means of graphical symbols, the electrical connections and functions of a specific circuit arrangement.
15)	Wiring and cable harness drawings —documents the path of a group of wires laced together in a specific configuration, so formed to simplify installation.
16)	Models, simulations, or design databases —provides a physical, analytical, or digital representation of any of the items listed above.
B. Software	
1)	Software design documentation —documents the software items architecture, design requirements, implementation logic, and data structures that provide a means of support.
2)	Software source code listings —documents the actual source code instructions that represented the “as-built” implementation.
C. Human —documents cognitive, physical, and sensory characteristics of the humans who operate, maintain, and support the system throughout its life cycle that directly contribute to, or constrain, system performance and impact human-machine interfaces.	
1)	Manpower, personnel, and training documentation —documents the knowledge, skills, and abilities; training requirements; and availability of the humans who operate, maintain, and support the system throughout its life cycle.
2)	Workspace arrangement drawings —documents the relationship of humans that support each phase of the life cycle with major subsystems or components of the system.
3)	Interface design specifications and drawings —documents all interfaces between the humans that support the system life cycle and any aspect of the system, including human-human interfaces, human-hardware interfaces, and human-software interfaces.
4)	Operational sequence diagrams —a graphical representation that documents the interaction between humans and other subsystems or components of the system in the performance of a task over time.
5)	Procedures —documents the actions that humans who support each phase of the life cycle perform to develop, produce, test, distribute, operate, support, and dispose of the system or its products, or to train humans to accomplish these actions. These procedures may be in the form of operational sequence diagrams, lists, or tables.
6)	Safety specifications —documents the equipment/system design features, performance specifications, and training that reduces the potential for human or machine errors or failures that cause injury or death within the constraints of operational effectiveness, time, and cost throughout the equipment/system life cycle.
D. Life cycle processes	
	Process product design architecture —documents the design architecture for the life cycle process products related to development (systems engineering and integration), manufacturing, verification, distribution, support, training, and disposal. Products include equipment, software, people, facilities, processes, and services integral to a specific life cycle process.

4.7.1 Hardware

The integrated data package shall include the technical design information that supports product manufacturing, assembly, and integration.

4.7.2 Software

The integrated data package includes the technical design information that supports software requirements, design, source code, verification and validation, build instructions, operation, and maintenance.

4.7.3 Life cycle processes

The integrated data package includes the technical design information for the life cycle processes, and the special equipment specifications and baselines, software code listings, technical manuals, technical plans, facility drawings, and special tools related to manufacturing, verification, distribution, operations, support, training, and disposal.

4.7.4 Human

The integrated data package includes the information that supports the definition of the role of the humans who support the system life cycle. Example products include the human systems engineering plan, safety specifications, manpower, personnel, and training documentation; workspace arrangement drawings; interface design specifications and drawings; operational sequence diagrams; manual procedures; and models, simulations, or design databases.

4.8 Specification tree

The enterprise shall generate a specification tree modeled after the design architecture appropriate to the level of development. The specification tree is composed of specification elements and interface specifications. Interface specifications document the interface requirements among interacting elements. System interface specifications define interfaces with external systems, platforms, and products. Subsystem interface specifications define interfaces among subsystems, including hardware-hardware, hardware-software, software-software, human-human, human-hardware, and human-software interfaces. The various specifications that define the elements of a fully developed system are shown in Figure 5. Since enterprise generally works on an element within a higher-level system, Figure 5 reflects how such an effort could be viewed in a system context. The specification tree is developed top down. The number of levels of a specification tree depends on the stage of the life cycle, as described in Clause 5. The lowest level of the specification tree is dependent upon the complexity of the design element and to what level the decision can be made to make, buy, or reuse the product associated with the specification. In the case of the humans who support the system throughout its life cycle, the decision is whether personnel with the appropriate knowledge, skills, and abilities are available, or what level of hiring, recruiting, or training needs to be performed. This could be at the subsystem level or any level down to the component level.

4.9 Drawing tree

The enterprise shall generate a drawing tree to reflect the drawings associated with the elements of the design architecture. This tree should resemble the specification tree of Figure 5.

4.10 System breakdown structure (SBS)

The enterprise shall generate an SBS to depict the hierarchy of products and processes that comprise the system architecture. The SBS shall be used by an enterprise for development and control of planning packages and their conversion to work packages, sizing of work packages, configuration management, resource allocation, traceability of requirement changes, interface control, cost reporting, and event-based scheduling. The life cycle processes are assigned to the product, assembly, and, if needed, the subassembly level of the SBS

and the humans who support the life cycle. The manpower, personnel, and training specifications that define the knowledge, skills, abilities, availability, and training required for the humans who support the system, its subsystems, and components are developed to ensure that the system may be adequately operated and supported throughout its life cycle. The life cycle processes may be defined at each level of the SBS, as necessary, to provide life cycle sustainment for elements of the product. The various system elements and life cycle processes that make up an SBS for a representative fully developed system are shown in Figure 5. This figure reflects the three major levels of development of a system—system definition, preliminary design, and detailed design—through which a system evolves, as described in Clause 5.

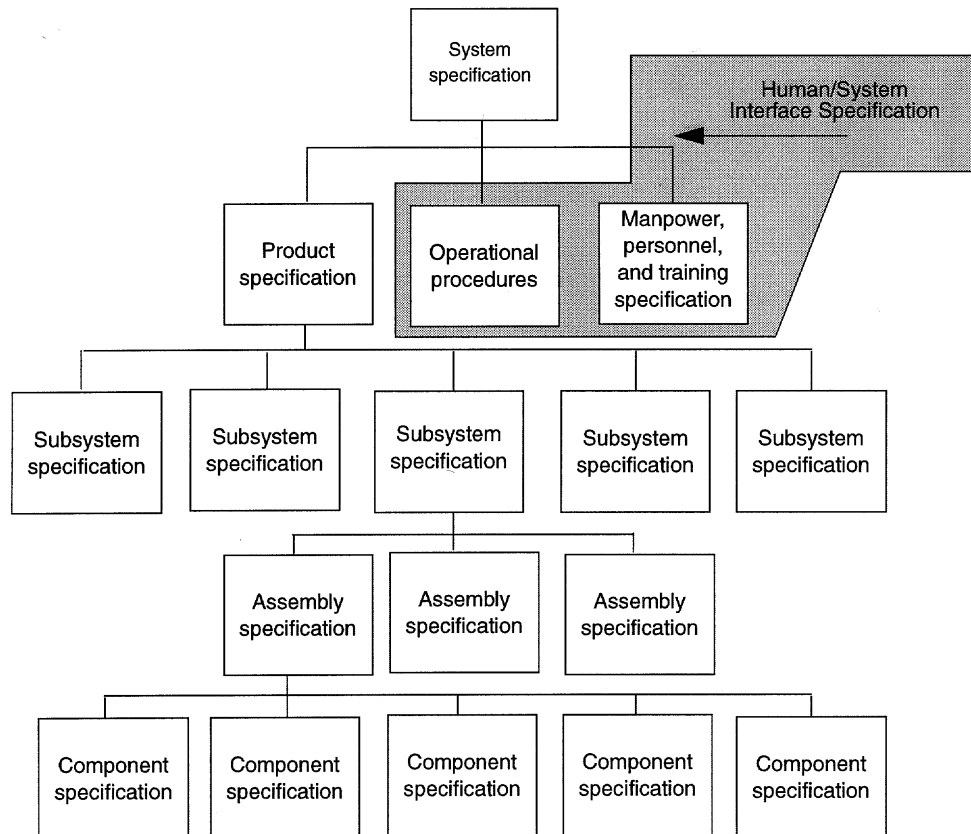


Figure 5—The specification tree

4.11 Integration of the systems engineering effort

The enterprise shall integrate the various inputs of the engineering and business specialties into the systems engineering effort to meet project objectives.

4.11.1 Concurrent engineering

The enterprise shall strive to integrate the concurrent design of products and their related life cycle processes. Concurrent engineering should integrate product and process development to ensure that the product(s) are producible, usable, and supportable. Computer-aided engineering tools should be used to support development and manufacturing, and should be integrated into the integrated database.

4.11.2 Integrated teams

Integrated teams should be considered as a primary approach to organizing for efficiency and effectiveness. When integrated teams are utilized, the enterprise should assign teams to specific elements of the SBS. Each team should prepare the required planning documents (engineering plan, master schedule, etc.) for the sys-

tem element to which it is assigned; should be responsible for developing and satisfying the specifications and baselines associated with the element; and should complete the work outlined in tasking statements related to the element, including the technical reviews of Clause 5.

4.12 Technical reviews

The enterprise shall conduct reviews, to include design reviews (system, subsystem, component, life cycle processes, test readiness, production approval) and audits (functional and design configuration), for the purpose of assessing technical progress. Normally, a design review should be conducted at the completion of each application of the SEP. Each review should accomplish the following:

- a) Assess the system requirements and allocations to ensure that requirements are unambiguous, consistent, complete, feasible, verifiable, and traceable to top-level system requirements;
- b) Assess the design maturity based on technical development goals, SEMS events and accomplishments, and empirical analysis and test data supporting progress to date;
- c) Present the risks associated with a continued development effort;
- d) Assess the life cycle processes and infrastructure necessary for product sustainment throughout the system life cycle;
- e) Identify resources required for continued development;
- f) Determine whether to proceed with the next application of the SEP, to discontinue development, or to take corrective actions before proceeding with the development effort.

Component, subsystem, and system design reviews shall be conducted, as appropriate, for each level of development. Depending on the complexity of the system, lower-level reviews may be needed (see Figure 6). Trade-off analysis and verification results should be available during design reviews in order to substantiate design decisions. Reviews may result in the need to iterate through the SEP to resolve identified deficiencies before progressing further in the development activity. Component, subsystem, and system functional- and design-configuration audits shall be performed to ensure that supporting documentation has been satisfactorily completed, that qualification tests for each specification requirement have been completed and all requirements satisfied, and/or that products comply with final drawings.

4.13 Quality management

The enterprise shall document and apply quality-management procedures for the development of products and life cycle processes consistent with established quality practices (see IEEE/EIA 12207.0-1996 [B2]).

4.14 Product and process improvement

The enterprise shall monitor product and process quality factors and shall continuously improve products and processes throughout the system life cycle, as described in the following subclauses.

4.14.1 Re-engineering

The enterprise should explore ways to improve existing products and processes. The enterprise should capture the design data, schema, tools, and models related to each development effort and design in the capacity to evolve a product or process in order to improve system cost effectiveness and to correct deficiencies.

4.14.2 Self assessment

The enterprise shall maintain a self-assessment program to determine the maturity of its systems engineering practices to include the application of the SEP tasks of Clause 6. The enterprise applies insights gained during self-assessment toward improving products, life cycle processes, and enterprise systems engineering practices.

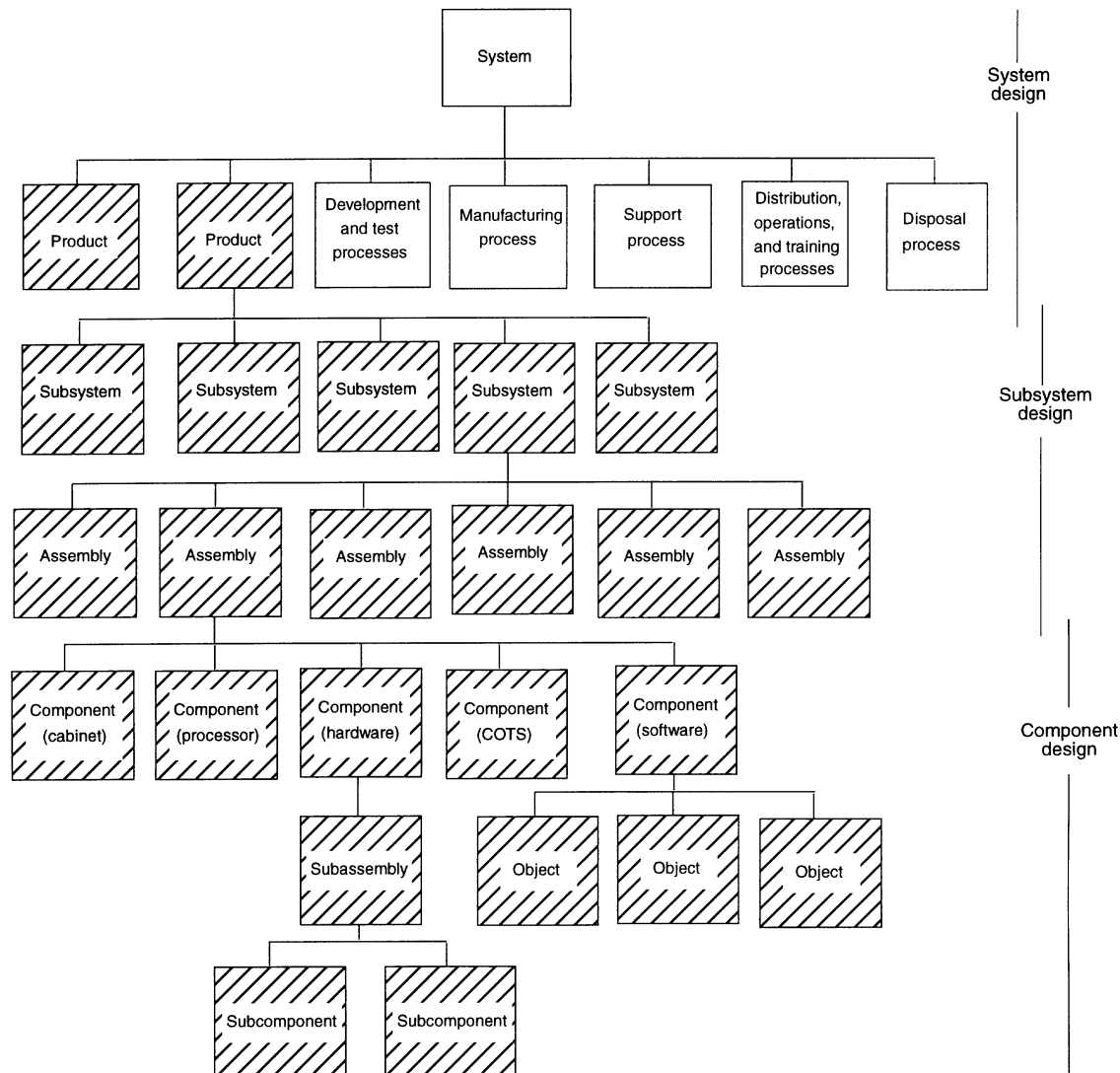


Figure 6—The system breakdown structure (SBS)

4.14.3 Lessons learned

The enterprise shall capture the lessons learned on each project and incorporate them into enterprise training courses, as appropriate, to improve the application of the SEP. Lessons learned provide a basis for establishing future system development efforts, for improving metrics, and for avoiding problems encountered by previous projects undertaken by the enterprise.

5. Application of systems engineering throughout the system life cycle

The enterprise shall apply the SEP (described in Clause 6) throughout the system life cycle for development and support of system products and their life cycle processes related to development, manufacturing, verification, distribution, support, training, and disposal. The SEP shall be applied during each level of system development (system, subsystem, and component) to add value (additional detail) to the products defined in the prior application of the process. The SEP shall be applied during fabrication, assembly, integration, and test (FAIT), production, and consumer support stages to resolve reported problems, and for evolving prod-

ucts to improve performance or extend service life. The typical system life cycle stages of development and operations are as follows:

Stages of development

- a) System definition;
- b) Subsystem definition
 - 1) Preliminary design of subsystems;
 - 2) Detailed design of subsystem components;
 - 3) Fabrication, assembly, integration, and test;

Stages of operations

- a) Production;
- b) Customer support.

These stages are depicted in Figure 7.

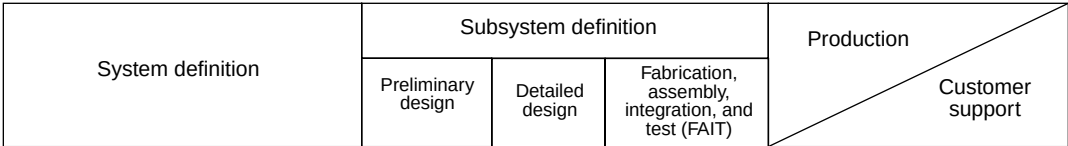


Figure 7— Typical system life cycle

The enterprise should satisfy the major exit events for each development stage of the system life cycle as defined in 5.1 through 5.3. These major events include establishing product descriptions, completing specifications, establishing configuration baselines, and completing technical reviews. The enterprise should also accomplish the systems engineering tasks described in 5.4 and 5.5, which include correcting design and process deficiencies/problems, evolving the system to provide added capabilities, extending the life of the system, and completing technical reviews. Additionally, the enterprise should develop the life cycle processes that are needed for system products to satisfy total life cycle needs and requirements. The activities related to life cycle process developments are discussed in 5.6.

5.1 System definition stage

The enterprise shall execute the system definition stage to establish the definition of the system with a focus on system products required to satisfy operational requirements. The major events of this stage should include completion of system, product, and subsystem interface specifications, system and product specifications, and preliminary subsystem specifications; establishment of a system baseline; and completion of technical reviews appropriate to the system definition stage. The documentation produced during system definition is required to guide subsystem developments. The technical reviews should evaluate the maturity of the system development and the readiness to progress to subsystem definition.

5.1.1 System definition

The enterprise shall apply the SEP, described in Clause 6, for the purpose of generating system-level validated requirements baseline, verified functional and design architectures, specifications and system baseline, SBS, and updated engineering and technical plans. Specific activities to be accomplished are listed in Table 2.

5.1.1.1 System concept

For precedented system developments, with incrementally improved or evolutionary growth products, the enterprise should refine established system definitions and product-line strategies to satisfy the market opportunity or customer order. For unprecedented systems where the concept is not already defined, the