

This is an **Archive IEEE Standard**. It has been withdrawn or superseded by a later version of the standard.

ARCHIVE STANDARDS MAY HAVE VALUE AS HISTORICAL DOCUMENTS, BUT THEY ARE NOT VALID OR APPROVED IEEE STANDARDS. THE IEEE SHALL NOT BE LIABLE FOR ANY DAMAGES RESULTING FROM THE SUBSCRIBER'S OR USERS' PRACTICE OF THIS UNAPPROVED, ARCHIVE IEEE STANDARD.

For more information about this or other current, revised or withdrawn standards, please consult the IEEE Standards Status report on-line at <http://standards.ieee.org/db/status/>.

IEEE Trial-Use Standard for Application and Management of the Systems Engineering Process

Sponsor

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

Approved December 13, 1994

IEEE Standards Board

Abstract: The interdisciplinary tasks that are required throughout a system's life cycle to transform customer needs, requirements, and constraints into a system solution are defined. This standard applies to a performing activity within an enterprise that is responsible for developing a product design and establishing the life-cycle infrastructure needed to provide for life-cycle sustainment. It specifies the requirements for the systems engineering process and its application throughout the product life cycle. The requirements of this standard are applicable to new products as well as incremental enhancements to existing products.

Keywords: enterprise, systems engineering detailed schedule (SEDS), systems engineering management plan (SEMP), systems engineering master schedule (SEMS), systems engineering process

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

Copyright © 1995 by the Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 1995. Printed in the United States of America.

ISBN 1-55937-496-9

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 Technical Committees of the IEEE Societies and the Standards Coordinating Committees of the IEEE 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 technical 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 Standards Board
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
USA

IEEE Standards documents may involve the use of patented technology. Their approval by the Institute of Electrical and Electronics Engineers does not mean that using such technology for the purpose of conforming to such standards is authorized by the patent owner. It is the obligation of the user of such technology to obtain all necessary permissions.

Publication of this trial-use standard for comment and criticism has been approved by the Institute of Electrical and Electronics Engineers. Trial-use standards are effective for 24 months from the date of publication. Comments for revision will be accepted for 18 months after publication. Suggestions for revision should be directed to the Secretary, IEEE Standards Board, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08817-1331, and should be received no later than August 15, 1996. It is expected that following the 24-month period, this trial-use standard, revised as necessary, shall be submitted to the IEEE Standards Board for approval as a full-use standard.

Introduction

(This introduction is not a part of IEEE Std 1220-1994, IEEE Trial-Use Standard for Application and Management of the Systems Engineering Process.)

History of the standard

The project authorization request for this standard was approved by the IEEE Standards Board in August 1989. The working group was organized and conducted its first meeting in January 1990. From the period of May 1990 until July 1993 a significant effort was expended to coordinate with a parallel standardization effort within the Department of Defense in preparation of MIL-STD-499B. The intent of the close coordination between the Military standardization project and the IEEE P1220 project was to ensure that the standards represented a consistent view of Systems Engineering. In July, 1993 the IEEE P1220 Working Group focused on issuing the standard for balloting. The balloting process was initiated in late December 1993, and was closed out in June 1994.

Because this standard is recognized as being the first commercial standard to address the Application and Management of the Systems Engineering Process this standard will be issued as a trial-use standard. It is anticipated that a joint effort will be undertaken over the next two years to merge this standard and the commercial version of MIL-STD-499B currently being prepared by a joint working group of CODSIA (Council of Defense and Space Industry Associations), NCOSE (National Council on Systems Engineering), and government members. It is anticipated that the commercial version of MIL-STD-499B will be published by the EIA (Electronic Industry Association) as an interim standard.

Objective of this standard

This standard defines the requirements for an enterprise's total technical effort related to development of consumer products (including computers and software) and processes that will provide life-cycle support. It prescribes an integrated technical approach to engineering a system and requires the application and management of the systems engineering process throughout a typical product life cycle. This standard defines the disciplined systems engineering process that is to be applied and managed during a systems engineering effort. This process is applied recursively to the development or incremental improvement of a consumer product to satisfy market requirements and to simultaneously provide related life-cycle processes for product development, manufacturing, test, distribution, 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 system engineer is responsible. Rather it encompasses what all organizational entities and all enterprise personnel must accomplish to produce a quality, competitive consumer 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 correct performance features at an affordable price and on time. This involves the establishment of an integrated set of products (hardware, software, data, facilities, and material) and processes (services and techniques) that are acceptable to customers and satisfy enterprise and external constraints, and considers and defines the role of people in developing, producing, testing, handling, operating, and supporting the system's products and 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.

It is not within the scope of this standard to address the roles, responsibilities, training requirements, or related aspects associated with an enterprise's implementation of systems engineering. Such topics provide the "how to implement" details, and thus, are best left to the enterprise for implementation. This standard solely addresses "what" the systems engineering process is, and how it must be applied and managed throughout the system life cycle.

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

Richard Schmidt, *Chair*

Jim Armstrong
Dr. George Friedman

Bill Gess
Jim Lacy
Dr. Jerry Lake

Ken Ptack, *Editor*

Joel O'Rourke
John Snoderly

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

Jim Cloud

Roger D. Larson
Dr. Arnold M. Ruskin

Jack Sivak

Other individuals who have contributed review and comments are:

HuCheng
Peter Eirich
Gregg Giesler
April Gillam

Raamzi Haraty
Diane Kang
Ed Kulas
H. Reiche
David Schultz

J.J. Selis
Jim Swanek
L. Mark Walker
Peter F. Zoll

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

AIA
ASC X3
ASQC

Department of Defense
EIA
ISO/IEC/JTC1/SC7
SSC10

MSC
NNCC

The following persons were on the balloting committee:

Frank Ackerman	Julio Gonzalez Sanz	David Peercy
Mark Amaya	Donald Gotterbarn	Alfred Perschel
Motoei Azuma	Michael Guba	Peter Poon
Edward Bartlett	David Gustafson	Robert Poston
Oddur Benediktsson	Jon Hagar	Kenneth Ptack
H. R. Berlack	Walter Heimerdinger	Ali Rahimi
Richard Biehl	Manfred Hein	Meir Razy
Mark Bilger	John Horch	John Reddan
William Boll, Jr.	Garland Jett	John Riganati
Sandro Bologna	David Johnson	Gregory Russell
Ronald Braun	Frank Jorgensen	William Sabor
Bruce Brocka	Laurel Kaleda	Kenneth Salter
Fletcher Buckley	Fumimiko Kamijo	Ronald Saltgaver
Susan Burgess	Eiichi Kaneko	Douglas Schaus
Juan Cacho	Richard Karcich	Peter Schilling
Jaya Carl	John Kellerman	Richard Schmidt
Neva Carlson	Ron Kenett	Norman Schneidewind
Leslie Chambers	Judy Kerner	Gregory Schumacher
Basilio Chen	Scotty Kilbourne	Leonard Seagren
John Chihonek	F. L. King	Carl Seddio
Tsun Chow	Peter Klopfenstein	Terry Shepard
François Coallier	Dwayne Knirk	Keith Shewbridge
Lee Cooper	Shaye Koenig	Robert Shillato
G. M. Cornefert	Joan Kundig	Katsutoshi Shintani
Geoff Cozens	Lak Ming Lam	David Siefert
Taz Daughtrey	John Lane	Ronald Skoog
Paul Davis	Boniface Lau	Richard Sky
Felix De Onandia	Dennis Lawrence	Michael Smith
Bostjan Derganc	Bertil Lindberg	Harry Sneed
Rodney Dorville	Michael Lines	Alfred Sorkowitz
Einar Dragstedt	William Lively	Vijaya Srivastava
Scott Duncan	Ben Livson	Richard Staunton
Mokhtar El Asaad	James Longbucco	Richard Stevens
Caroline Evans	Joseph Maayan	Richard Thayer
Richard Evans	Jukka Marijarvi	Douglas Thiele
Richard Farley	James W. McClean	Leonard Tripp
John Fendrich	Patrick McCray	Margaret Updike
Peter Fillery	Russell McDowell	Ted Urbanowcz
Jay Forster	Sue McGrath	Glenn Venables
Kirby Fortenberry	Christopher McMacken	Peter Voldner
Richard Fries	Glen Meldrum	Ronald Wade
Eitan Froumine	James Miller	Dolores Wallace
Simon Gabrihelidis	Louis Miller	John Walz
Barry Garner	Rajko Milovanovic	Camille White-Partain
Yair Gershkovich	Celia Modell	Paul Work
Adel Ghannam	Hironobu Nagano	F. D. Wright
Shirley Gloss-Soler	Dennis E. Nickle	Natalie Yopconka
John Glynn	Michael O'Neill	Weider Yu
Rajeev Goel	Mike Ottewill	Janusz Zalewski
	Robert Parys	

When the IEEE Standards Board approved this standard on December 13, 1994, it had the following membership:

Wallace S. Read, *Chair*

Donald C. Loughry, *Vice Chair*

Andrew G. Salem, *Secretary*

Gilles A. Baril
Bruce B. Barrow
José A. Berrios de la Paz
Clyde R. Camp
James Costantino
Stephen L. Diamond
Donald C. Fleckenstein
Jay Forster*
Ramiro Garcia

Donald N. Heirman
Richard J. Holleman
Jim Isaak
Ben C. Johnson
Sonny Kasturi
Lorraine C. Kevra
E. G. "Al" Kiener
Ivor N. Knight

Joseph L. Koepfinger*
D. N. "Jim" Logothetis
L. Bruce McClung
Marco W. Migliaro
Mary Lou Padgett
Arthur K. Reilly
Ronald H. Reimer
Gary S. Robinson
Leonard L. Tripp

*Member Emeritus

Also included are the following nonvoting IEEE Standards Board liaisons:

Satish K. Aggarwal
James Beall
Richard B. Engelman
Robert E. Hebner

Rachel A. Meisel
IEEE Standards Project Editor

CONTENTS

CLAUSE

1.	Overview.....	1
1.1	Scope.....	1
1.2	Purpose.....	2
1.3	Understanding this standard.....	2
1.4	Organization of this standard.....	4
2.	References.....	5
3.	Definitions and acronyms	6
3.1	Definitions.....	6
3.2	Acronyms.....	11
4.	General requirements.....	12
4.1	Systems engineering process	12
4.2	Policies and procedures for systems engineering	13
4.3	Planning the technical effort.....	13
4.4	Evolutionary development strategies.....	14
4.5	Modeling and prototyping.....	14
4.6	Integrated database.....	14
4.7	Product and process data package.....	15
4.8	Specification tree	15
4.9	Drawing tree.....	15
4.10	System breakdown structure (SBS)	17
4.11	Integration of the systems engineering effort	17
4.12	Technical reviews	18
4.13	Quality management.....	19
4.14	Continuing product and process improvement	20
5.	Application of systems engineering throughout the system life cycle	21
5.1	System definition stage.....	21
5.2	Preliminary design stage.....	25
5.3	Detailed design stage	29
5.4	Fabrication, Assembly, Integration, and Test (FAIT) stage	33
5.5	Production and customer support stages.....	35
5.6	Simultaneous engineering of products and services of life-cycle processes	37
6.	The systems engineering process.....	39
6.1	Requirements analysis	39
6.2	Requirements validation	44
6.3	Functional analysis.....	45
6.4	Functional verification.....	48
6.5	Synthesis	49
6.6	Physical verification.....	54
6.7	Systems analysis	57
6.8	Control	62

ANNEXES

A	(informative) The role of systems engineering within an enterprise.....	69
B	The Systems Engineering Management Plan (SEMP).....	73

IEEE Trial-Use 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. The standard is intended for guiding the development of systems (which includes computers and software) for commercial, government, military, and space applications. This standard applies to a performing activity 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 and its application throughout the product life cycle. 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, to products which 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 consumer product will be utilized, and for which the product must be designed to accommodate
- b) Defining the consumer product requirements in terms of functional and performance requirements, quality factors, producibility, supportability, safety, and environmental impacts
- c) Defining the life-cycle process products for manufacturing, test, distribution, support, training, and disposal, which are necessary to provide life-cycle support for consumer products

Given the project's authorization and objectives, the performing activity should establish an Systems Engineering Management Plan (SEMP), Systems Engineering Master Schedule (SEMS), and a Systems Engineering Detail Schedule (SEDS) that provide the planning and scheduling of technical efforts to accomplish project objectives. The SEMP is the main planning document for all project technical efforts and describes the tailored application of this standard. The SEMS, an event-based schedule, and the SEDS, a calendar-based schedule derived from the SEMS, address the development activities for the consumer product, as well as supporting life-cycle processes. Thus, within this standard, when the term *system* is used, it implies a total system concept that involves the consumer products and their supporting life-cycle processes.

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 more acute to engineer each of those products as a total system.

1.3 Understanding this standard

The description of the systems engineering process 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.

A system can be viewed as an element of a larger system. The challenge is to understand the boundary of the system that is the focus of the development effort as well as the relationships and interfaces among this system to other systems. Figure 1 depicts how a system may relate to other systems and higher-level systems. On a smaller scale, and the focus of this standard, are product-oriented systems such as the automobile, airplane, or computers.

A system is typically composed of several related elements and their interfaces. Each element may be composed of hardware, software, data, facilities, materials, services, and techniques. Additionally, an element has associated with it the people required to develop, produce, test, distribute, operate, support, or dispose of the element's products or train other people to accomplish their role in a system context. Figure 2 provides a hierarchy of terms for the elements making up a system. This generic system hierarchy is a key concept within this standard because it ties the physical and system architectures, specification and drawing trees, system breakdown structure, technical reviews, and configuration baselines together.

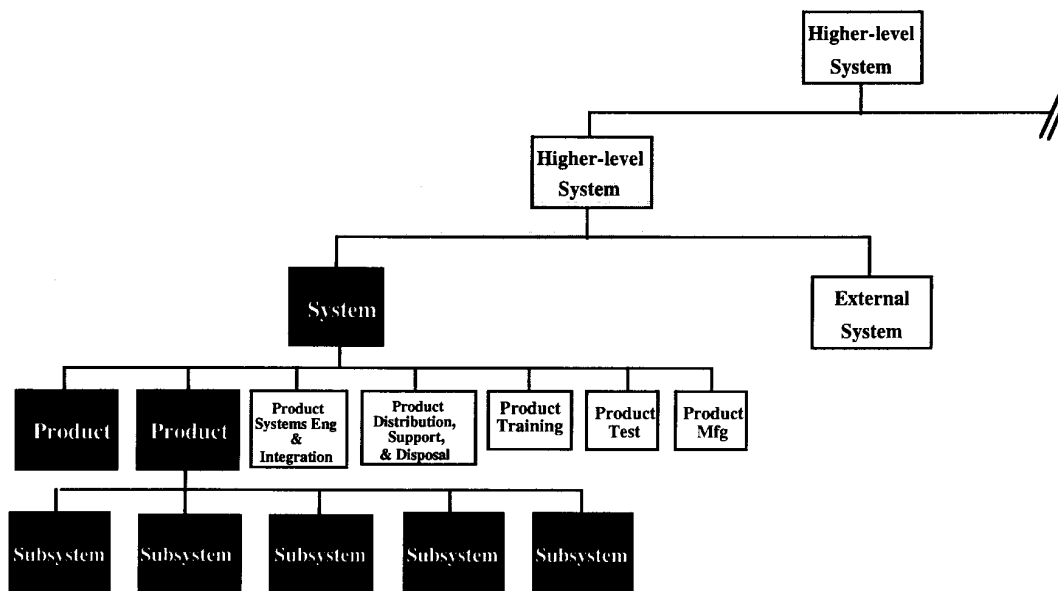


Figure 1 – Hierarchy of systems

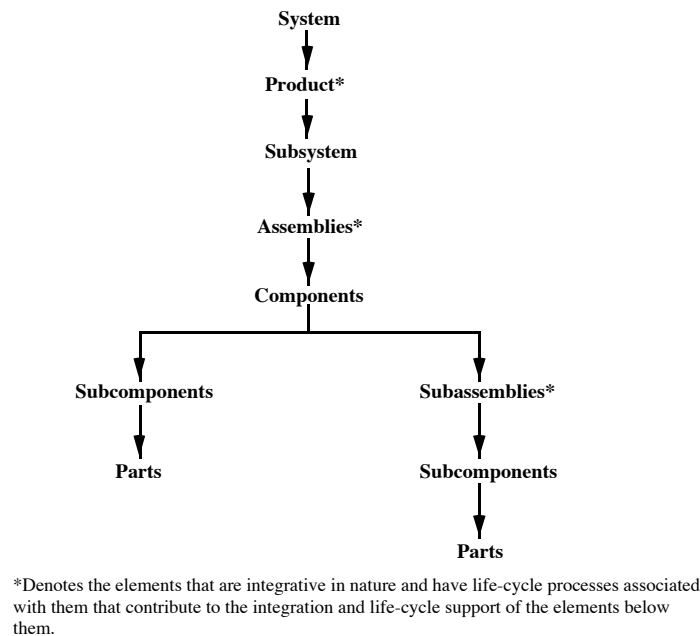


Figure 2—Hierarchy of elements within a system

From a development viewpoint, an actual system may consist of only the first three levels—System, Product, and Subsystems—or it can be composed of multiple levels of elements down to the parts level as shown in figure 2. The basic building blocks of a system—the system, its related product(s), the life-cycle processes required to support the products and customers for the product(s), and the subsystems that make up the product(s)—are depicted in figure 3. Each life-cycle process—development, manufacturing, test, distribution, support, training, and disposal—is itself like a system in that products must be developed to fulfill the purpose of the life-cycle process. For example, a product must be manufactured. Manufacturing is a life-cycle process. The products associated with the manufacturing life-cycle process include special equipments, tools, facilities, and production processes and procedures.

A typical system is composed of *products* produced by suppliers/subcontractors. Each supplier/subcontractor considers their *product(s)* as part of their *system*. The organization that purchases these *systems* for integration into a higher-level system will refer to these *systems* as subcomponents, components, or subsystems, depending on the significance of the element in contributing to the functionality of the systems or costs.

Figure 4 depicts the three levels of development associated with a total system development. These levels with their associated tasks and outputs are described in clause 5. Note that during system definition the system and its products are defined in specifications, the needs for the life-cycle processes are identified, and the subsystems that make up a product are identified and defined in preliminary configuration documents. During subsystem definition each subsystem is treated in the same manner as the system above. The subsystem and its assemblies are fully defined and the need for life-cycle processes are identified; and components that make up the assembly are identified and preliminary configuration documents completed. Likewise in component level design, each component may be looked upon as a system with its products (subassemblies) and associated life-cycle processes. Each subassembly may have several subcomponents. Component design may continue to be decomposed down to parts or may be stopped at any time at an element of the structure until a point is reached needing no further decomposition to be considered fully designed.

The definitions of system elements are generated by activities of the systems engineering process. This process is shown in figure 5 and described in detail in clause 6. The systems engineering process 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.

1.4 Organization of this standard

- *Clause 1* provides the scope, purpose, and organization of this standard.
- *Clause 2* lists the standards and acronyms applicable to this document.
- *Clause 3* establishes the meaning of terms as used in this standard.
- *Clause 4* establishes requirements for planning and implementing an effective systems engineering capability within an enterprise.
- *Clause 5* describes of the application of the systems engineering process through system definition, subsystem definition, production, and customer support.
- *Clause 6* details the tasks of the systems engineering process to be tailored and performed to develop consumer product solutions and their supporting life-cycle processes.
- *Annex A* discusses the systems engineering process 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.

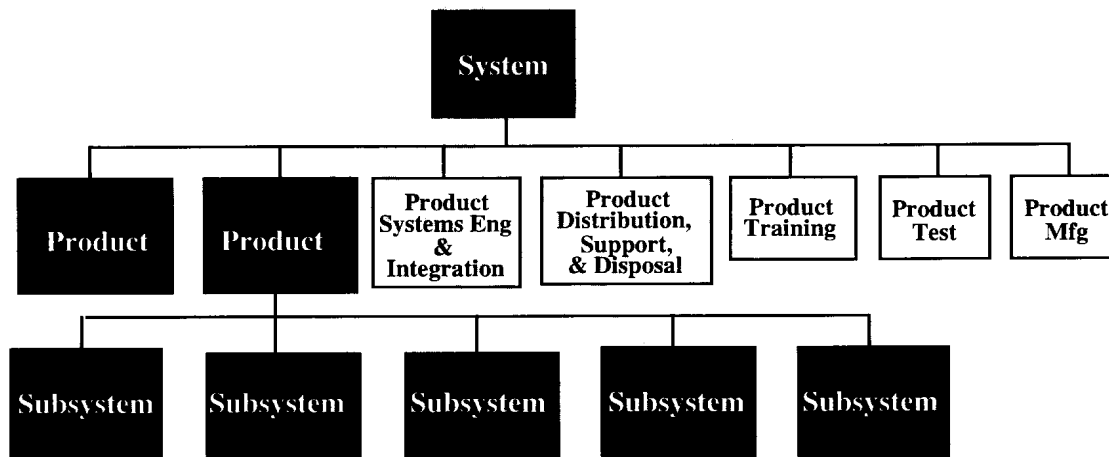


Figure 3— Basic building block of a system

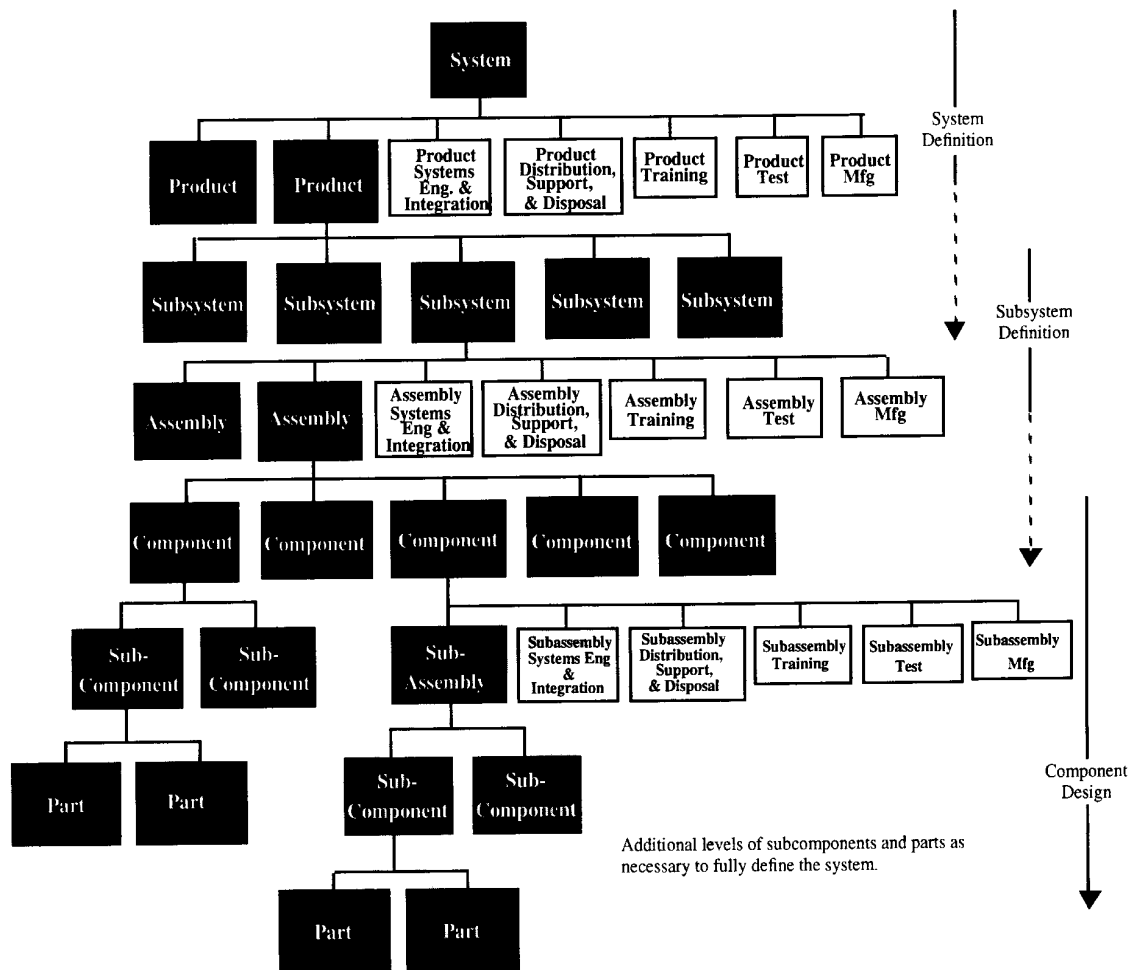


Figure 4—Development of the system

2. References

This standard shall be used in conjunction with the following standards:

ANSI/ASQC Q91-1987, Quality Systems—Model for Quality Assurance in Design/Development, Production, Installation, and Servicing.¹

IEEE Std 100-1992, IEEE Standard Dictionary of Electrical and Electronics Terms (ANSI).²

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

IEEE Std 1012-1986, IEEE Standard for Software Verification and Validation Plans (ANSI).

IEEE P1233 November 1993, Guide for Developing System Requirements Specifications.³

¹ANSI publications are available from the Sales Department, American National Standards Institute, 11 West 42nd Street, 13th Floor, New York, NY 10036, USA.

²IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, USA.

³This authorized standards project was not approved by the IEEE Standards Board at the time this went to press. It is available from the IEEE.

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 IEEE Std 100-1992⁴ and this standard, then the definitions provided here should take precedence as they apply only to the practices and requirements of this standard.

3.1.1 assembly (n): An element of the physical or system architecture, specification tree, and system break-down structure that is a subordinate element to a subsystem and is comprised of two or more components. It represents a consumer product (automatic brake system) of a subsystem (braking system of an automobile); or a life-cycle process product (control system) of a subsystem (flight controls of a simulator) related to a life-cycle process (training) that supports an assembly or group of assemblies.

3.1.2 assembly (v): The act of fitting together fabricated or manufactured elements into a larger element.

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

3.1.4 component: An element of the physical or system architecture, specification tree, and system break-down structure that is a subordinate element to an assembly and may be composed of two or more subcomponents, or parts; or one or more subassemblies and their associated life-cycle processes. In a noncomplex system, the component may be the lowest element.

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

3.1.6 consumer product: An end item delivered to a customer.

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

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

3.1.9 design (n): The results of the synthesis process that provide sufficient details, drawings, or other pertinent information for a physical or software element that permits further development, fabrication, assembly, and integration, or production of a product element.

3.1.10 design (v): The act of preparing drawings or other pertinent information for a physical or software element during synthesis within the systems engineering process.

3.1.11 design-to-cost: Establishes a cost goal that is equivalent to any other performance parameter.

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

⁴Information on references can be found in clause 2.

3.1.13 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.

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

3.1.15 element: A product, subsystem, assembly, component, subcomponent or subassembly, or part of a physical or system architecture, specification tree, or system breakdown structure, including the system itself (see figure 2).

3.1.16 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.

3.1.17 environment: The natural (weather, climate, ocean conditions, terrain, vegetation, dust, etc.) and induced (electromagnetic, interference, heat, vibration, etc.) conditions that constrain the design solutions for consumer products and their life-cycle processes.

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

3.1.19 external system interfaces: The system or product interfaces to other systems, platforms, or products that influence the design solutions for consumer products and their life-cycle processes.

3.1.20 function: A task, action, or activity expressed as a verb-noun combination (e.g., Brake Function: stop vehicle) to achieve a defined outcome.

3.1.21 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.

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

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

3.1.24 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.

3.1.25 integration: The merger or combining of two or more lower-level elements into a functioning and unified higher-level element with the functional and physical interfaces satisfied.

3.1.26 interface specification: The description of essential functional, performance, and physical requirements and constraints at a common boundary between two or more system elements or between a system element and an operator (human interface).

3.1.27 life cycle: The system or product evolution initiated by a user need or by a perceived customer need through the disposal of consumer products and their life-cycle process products and by-products.

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

3.1.29 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 for the development of life-cycle process products and their life-cycle processes. The following are the eight essential functional processes.

- a) *Development.* The planning and execution of system and subsystem definition tasks required to evolve the system from customer needs to consumer 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 consumer 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 consumer product solutions and their life-cycle processes to measure specification compliance or customer satisfaction.
- d) *Distribution.* The tasks, actions, and activities to initially transport, receive, process, assemble, install, test, checkout, train, operate, and, as required, emplace, house, store, or distribute consumer products and life-cycle process products.
- e) *Operations.* The tasks, actions, and activities that are associated with the use of the consumer 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 tasks, actions, and activities to achieve and maintain the knowledge and skill levels necessary to efficiently and effectively perform operations, support, and disposal.
- h) *Disposal.* The tasks, actions, and activities to ensure that disposal or recycling of destroyed or irreparable consumer and life-cycle process products and by-products comply with applicable environmental regulations and directives.

3.1.30 life-cycle process product: An end item required to perform a life-cycle process in support of a consumer product. This end item may be a product, process, or service.

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

3.1.32 measure of performance (MOP): A performance measure that provides design requirements that are necessary to satisfy an MOE. There are generally several measures of performance for each measure of effectiveness.

3.1.33 method: A formal, well-documented approach for accomplishing a task, activity, or process step governed by decision rules to provide a description of the form or representation of the outputs.

3.1.34 mode: An operating condition of a function, subfunction, or physical element of the system.

3.1.35 operational conditions: The factors, including weather, human operations, external system interactions, etc. that contribute to defining operational scenarios or environments.

3.1.36 operational environment: The natural or induced environmental conditions, anticipated system interfaces, and user interactions within which the system is expected to be operated.

3.1.37 part: The lowest element of a physical or system architecture, specification tree, or system breakdown structure that can not be partitioned (e.g., bolt, nut, bracket, semiconductor, computer software unit).

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

3.1.39 performing activity: The person(s) or organization that performs the tasks specified in this standard.

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

3.1.41 physical characteristics: The physical *design* attributes or distinguishing features that pertain to a measurable description of a product or process.

3.1.42 physical element: A product, subsystem, assembly, component, subcomponent, subassembly, or part of the physical architecture defined by its designs, interfaces (internal and external), and requirements (functional, performance, constraints, and physical characteristics).

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

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

3.1.45 process: A sequence of tasks, actions, or activities, including the transition criteria for progressing from one to the next, that bring about a result.

3.1.46 product: An element of the physical or system architecture, specification tree, or system breakdown structure that is a subordinate element to the system and is comprised of two or more subsystems. It represents a major consumer product (e.g., automobile, airplane) of a system or a major life-cycle process product (e.g., simulator, building, robot) related to a life-cycle process that supports a product or group of products.

3.1.47 product and process data package: The evolving output of the systems engineering process that documents hardware designs, software designs with their associated documentation, and life-cycle processes.

3.1.48 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.

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

3.1.50 requirement: A statement that identifies a product or process capability, or physical characteristic that is unambiguous and quantitative whenever possible.

3.1.51 requirements baseline: The composite set of operational, functional, and physical requirements that serve to guide development and management decision processes.

3.1.52 requirements baseline validation: The process of evaluating the results of the requirements analysis activities of the systems engineering process to ensure compliance with customer expectations, project and enterprise constraints, and external constraints.

3.1.53 risk management: The activities associated with risk management preparation, risk assessment, risk handling option assessment, and risk control.

3.1.54 service: The delivery, installation, maintenance, training, and other labor-intensive activities providing life-cycle support associated with the products and processes of the system.

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

3.1.56 specification element: A product, subsystem, assembly, component, subcomponent, subassembly, or part of the specification tree described by a specification.

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

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

3.1.59 subassembly: An element of the physical or system architecture, specification tree, and system breakdown structure that is subordinate to a complex component, and is comprised of two or more subcomponents.

3.1.60 subcomponent: An element of the physical or system architecture, specification tree, and system breakdown structure that is subordinate to a noncomplex component, or a subassembly, and is comprised of two or more parts.

3.1.61 subsystem: An element of the physical or system architecture, specification tree, or system breakdown structure that is a subordinate element to a product and is comprised of one or more assemblies and their associated life-cycle processes.

3.1.62 system: The top element of the system architecture, specification tree, or system breakdown structure that is comprised of one or more products and associated life-cycle processes and their products and services.

3.1.63 system architecture: The composite of the physical architectures for consumer products and their life-cycle processes.

3.1.64 system breakdown structure (SBS): A hierarchy of elements and related life-cycle processes 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. The system breakdown structure is derived from the specification tree.

3.1.65 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; the reliability and maintainability of system elements; and the ability to produce, test, distribute, operate, support, train, and dispose of the system throughout its life cycle.

3.1.66 system element: A product, subsystem, assembly, component, subcomponent, subassembly, or part of the system breakdown structure that includes the specifications, configuration baseline, budget, schedule, and work tasks.

3.1.67 systems engineering: An interdisciplinary collaborative approach to derive, evolve, and verify a life-cycle balanced system solution that satisfies customer expectations and meets public acceptability.

3.1.68 tasking activity: The person(s) or organization that directs a performing activity to accomplish the work specified in this standard.

3.1.69 technical effort: The total engineering, testing, manufacturing, and specialty engineering effort associated with the development of a product that encompasses all of the system, equipment, facilities, etc., necessary for the enterprise to develop, produce, distribute, operate, test, support, train, and dispose of the product.

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

3.2 Acronyms

Term definitions are consistent with IEEE Std 610.12-1990.

ANSI	American National Standards Institute
BIT	Built-In Test
FAIT	Fabrication, Assembly, Integration, and Test
FIT	Fault-Isolation Test
FMEA	Failure, Modes, and Effects Analysis
IPT	Integrated Product Teams
MOE	Measure of Effectiveness
MOP	Measure of Performance
QFD	Quality Function Deployment
SBS	System Breakdown Structure
SEDS	Systems Engineering Detailed Schedule
SEMP	Systems Engineering Management Plan
SEMS	Systems Engineering Master Schedule
SPC	Statistical Process Control
TPM	Technical Performance Measurement

4. General requirements

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

- a) Plans, conducts, and manages a fully integrated technical effort necessary to satisfy the general requirements of this standard, as tailored for the specific project
- b) Applies the systems engineering process for each level of system decomposition (e.g., concept definition, system definition, subsystem design, fabrication, assembly, integration and test, production, and sustainment)
- c) Controls 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
- d) Generates models and prototypes to support trade-off analyses that support system architecting and design
- e) Generates a product and process data package which ensures that the product can be produced, tested, delivered, operated, supported, and properly disposed of
- f) Captures outputs from all technical activities in an integrated database

4.1 Systems engineering process

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

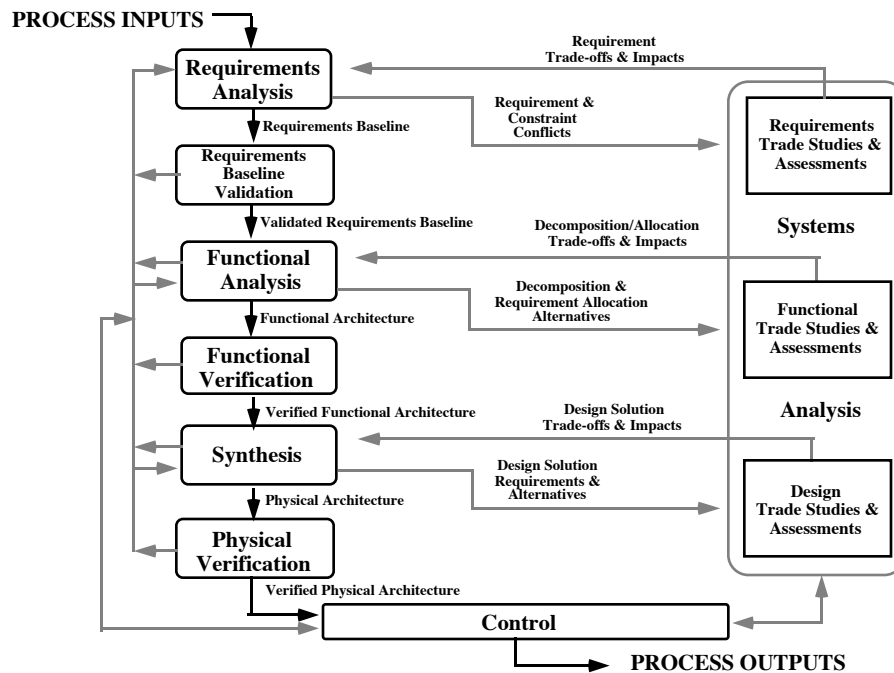


Figure 5—The system engineering process

4.2 Policies and procedures for systems engineering

The enterprise shall develop and maintain policies and procedures for governing the conduct of systems engineering within the enterprise. These policies and procedures specify requirements for the planning, implementation, and control of product development. These policies are selectively applied to varying business segments within the enterprise. Policy existence establishes the enterprise's paradigms for systems engineering, from which training can be based and project specific activities can be communicated. Enterprise policies and procedures shall address:

- Application of the systems engineering process throughout a project life-cycle
- Preparation and approval of a Systems Engineering Management Plan (SEMP)
- Preparation and approval of a Systems Engineering Master Schedule (SEMS) and Systems Engineering Detailed Schedule (SEDS)
- Preparation and approval of an evolving product and process data package
- Monitoring and reporting technical progress of the project
- Preparation for and conduct of technical and management reviews
- Contents and maintenance of an integrated database
- Continuous product and process improvement

4.3 Planning the technical effort

The performing activity shall prepare, implement, and maintain a Systems Engineering Management Plan (SEMP), Systems Engineering Master Schedule (SEMS), and Systems Engineering Detailed Schedule (SEDS) in accordance with the following subclauses.

4.3.1 Systems Engineering Management Plan (SEMP)

The SEMP guides and controls the technical efforts of the project. The SEMP reflects an integrated technical effort responsible for product development that balances all factors associated with meeting system life-cycle requirements. Annex B provides a generic outline and description of the contents for a SEMP. If an evolutionary or incremental development strategy is to be pursued, the SEMP includes an evolutionary development strategy for initial product development and insertion of incremental technology enhancements.

4.3.2 Systems Engineering Master Schedule (SEMS)

The SEMS provides a basis for event-based planning to establish key events, their significant tasks, and the criteria by which completion of significant tasks are determined. A properly designed SEMS allows for the scheduling of activities associated with tasks, loading of resources, budget preparation, assignment of personnel establishing task start and end dates, and event completion dates. The SEMS is structured to ensure that critical technical inputs and decision data are available for technical and project decision points and other identified events, and required progress and system maturity are demonstrated prior to continuing technical efforts dependent on that progress and maturity.

4.3.3 Systems Engineering Detailed Schedule (SEDS)

The SEDS provides a calendar-based schedule of activities, tasks, and key events of the SEMS. This calendar-based schedule is used to track progress and manage resources. SEDS data can 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 are prepared to supplement the SEMP as needed. Technical plans are prepared by the engineering and technical specialty areas to which they apply. They are used to implement activities and tasks and to measure technical progress against the plan. Technical plans are typically prepared for risk management, configuration management, technical reviews, verifications, computer resources, manufacturing, maintenance, training, security, and safety.

4.4 Evolutionary development strategies

The performing activity shall explore evolutionary development strategies for evolving the system and its capabilities. The capacity to change should be designed into the system architecture to enable the cost effective re-engineering of the system. This design attribute should be established early in the system life cycle to provide a basis for planning each incremental development effort. Evolutionary development strategies offer approaches for managing the introduction of new technologies, evolving requirements, or product capabilities.

4.5 Modeling and prototyping

The performing activity shall establish the models, simulations, or prototypes as required to 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 the system's functional and performance characteristics, producibility, supportability, environmental impact, safety, and human factors.

4.6 Integrated database

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

4.7 Product and process data package

The performing activity shall generate a product and process data package that documents architecture and design information for product manufacturing and life-cycle sustainment. The product and process data package includes the types of engineering data defined in table 1. The specific content of the product and process data package is defined early in the project life cycle, and shall consist of the classes of information identified in the subclauses below, and evolves with each application of the systems engineering process to provide more detail with each level of development.

4.7.1 Hardware

The product and process data package includes the technical design information that supports product manufacturing, assembly, and integration.

4.7.2 Software

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

4.7.3 Life-cycle processes

The product and process data package includes descriptions of the life-cycle process products related to development, manufacturing, verification, distribution, operations, support, training, and disposal. Example products include special equipment specifications and baselines, software code listings, technical manuals, technical plans, facility drawings, and special tools.

4.8 Specification tree

The performing activity shall generate a specification tree modeled after the verified physical 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 physical elements. System interface specifications define interfaces with external systems, platforms, and products. Subsystem interface specifications define interfaces among subsystems. The various specifications that define the elements of a fully developed system are shown in figure 6. Since a performing activity is generally working on an element within a higher-level system, figure 6 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 physical element and to what level the decision can be made to make, buy, or reuse the product associated with the specification. This could be at the subsystem level or any level down to the part level.

4.9 Drawing tree

The performing activity shall generate a drawing tree to reflect the drawings associated with the elements of the physical architecture. This tree should resemble the specification tree of figure 6.

Table 1—Product and process data package content^a

	Content	Purpose
<i>Hardware</i>	Arrangement drawings	Documents the relationship of the major subsystems or components of the system.
	Assembly drawings	Documents the relationship of a combination of parts and subassemblies required to form the next higher indenture level of equipment or system.
	Connection drawings	Documents the electrical connections of an installation or of its component devices or parts.
	Construction drawings	Documents the design of buildings or structures.
	Product drawings	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.
	Detail drawings	Documents complete end item requirements for the subcomponent(s) delineated in the drawing.
	Elevation drawings	Documents vertical projections of buildings and structures or profiles of equipment.
	Engineering drawings	Discloses by means of pictorial or textual presentations, or a combination of both, the physical and functional end product requirements or design of an item.
	Installation drawings	Documents general configuration and complete information necessary to install an item relative to its supporting structure or to associated items.
	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.
	Numerical control drawings	Documents complete physical and functional engineering and product requirements of an item to facilitate production by tape control means.
	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.
	Wire lists	Documents a book-form drawing consisting of tabular data and instructions required to establish wiring connections within or between items.
	Schematic diagrams	Documents, by means of graphical symbols, the electrical connections and functions of a specific circuit arrangement.
	Wiring and cable harness drawings	Documents the path of a group of wires laced together in a specific configuration, so formed to simplify installation.
	Models, simulations, or design databases	Provides a physical, analytical, or digital representation of any of the items listed above.

Table 1—Product and process data package content^a (Continued)

	Content	Purpose
<i>Software</i>	Software design documentation	Documents the software items architecture, design requirements, implementation logic, and data structures that provide a means of support.
	Software source code listings	Documents the actual source code instructions that represented the “as-built” implementation.
<i>Life-cycle processes</i>	Process product physical architecture	Documents the physical 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, facilities, processes, and services integral to a specific life-cycle process.

^aA Product and Process Data Package includes, but is not limited to, the items listed in this table.

4.10 System breakdown structure (SBS)

The performing activity shall generate a system breakdown structure modeled after the specification tree (see figure 6). The SBS is used by a performing activity for development and control of work packages, development of planning packages and their conversion to work packages, sizing of work packages, configuration management, resource use, technical reviews, traceability of requirement changes, interface control, cost reporting, and event-based scheduling. The system breakdown structure is a hierarchy of system elements and life-cycle processes. Each system element is described by the specification, configuration baseline, budget, work tasks, and schedules related to the element. The life-cycle processes are assigned to the product, assembly, and, if needed, the subassembly level of the SBS. The SBS is developed top down. The various system elements and life-cycle processes that make up an SBS for a representative fully developed system are shown in figure 7. This figure reflects the three major levels of system development—system definition, preliminary design, and detailed design—through which a system evolves as described in clause 5. It also shows the technical reviews typically associated with levels of development. The lowest level of the SBS is dependent on the lowest levels appropriate for the specification tree.

4.11 Integration of the systems engineering effort

The performing activity 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 should strive to integrate the concurrent design of products and their related processes through the use of an integrated, concurrent engineering environment. Other names for concurrent engineering are team design, simultaneous engineering, concurrent design, or integrated product and process development. Concurrent engineering integrates product and process requirements, organizes a project for efficiency and effectiveness, balances the project’s communications infrastructure, and integrates the system documentation infrastructure. Computer-aided engineering tools are used to support development and manufacturing within the concurrent engineering environment. These tools rely on an integrated data repository that contains the system requirements, design information, drawings, etc., as necessary to support the development and production of products.

4.11.2 Integrated teams

Integrated teams are considered as a primary approach to organizing for efficiency and effectiveness. When integrated teams are utilized, the project should assign teams to specific elements of the SBS. Each team

prepares required planning documents (SEMP, SEMS, etc.) for the system element to which assigned; is responsible for developing and satisfying the specifications and baselines associated with the element; and completes work outlined in tasking statements related to the element, including the technical reviews of clause 5.

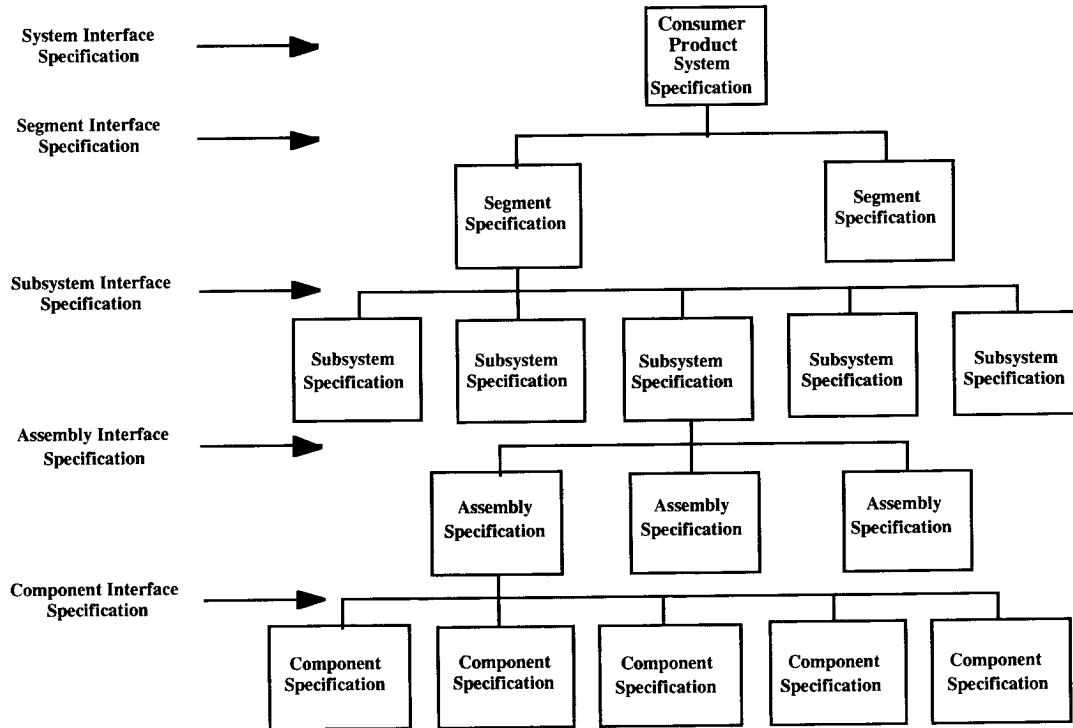


Figure 6—The specification tree

4.12 Technical reviews

The performing activity shall conduct technical reviews, to include design reviews (system, subsystem, component, life-cycle processes, test readiness, production approval) and audits (functional and physical configuration), for the purpose of assessing technical progress. Normally, a design review is held at the completion of each application of the systems engineering process.

A review does the following:

- Assesses the system requirements and allocations to ensure that requirements are unambiguous, consistent, complete, feasible, verifiable, and traceable to top-level system requirements
- Assesses the design maturity based on technical development goals, systems engineering master schedule events and accomplishments, and empirical analysis and test data supporting progress to date
- Presents the risks associated with a continued development effort
- Identifies resources required for continued development
- Decides whether to proceed with the next application of the systems engineering process, discontinue development, or take corrective action on the products and/or process of the current application before proceeding to the next application

Component, subsystem, and system design reviews are conducted as appropriate for each level of development. Depending on the complexity of the system, lower-level reviews may be needed (see figure 7). Life-cycle process reviews are held for each subassembly, assembly, or product, as appropriate. Trade study and verification results should be available during design reviews in order to substantiate design decisions. Component, subsystem, and system functional configuration audits and physical configuration audits are performed to ensure that tasks, activities, key events, and supporting documentation have been satisfactorily completed; that qualification tests for each specification requirement have been completed and all requirements satisfied; and/or that produced products comply with final drawings.

4.13 Quality management

The systems engineering process shall be documented and applied to develop products and life-cycle processes consistent with established quality practices (see ANSI/ASQC Q91-1987).

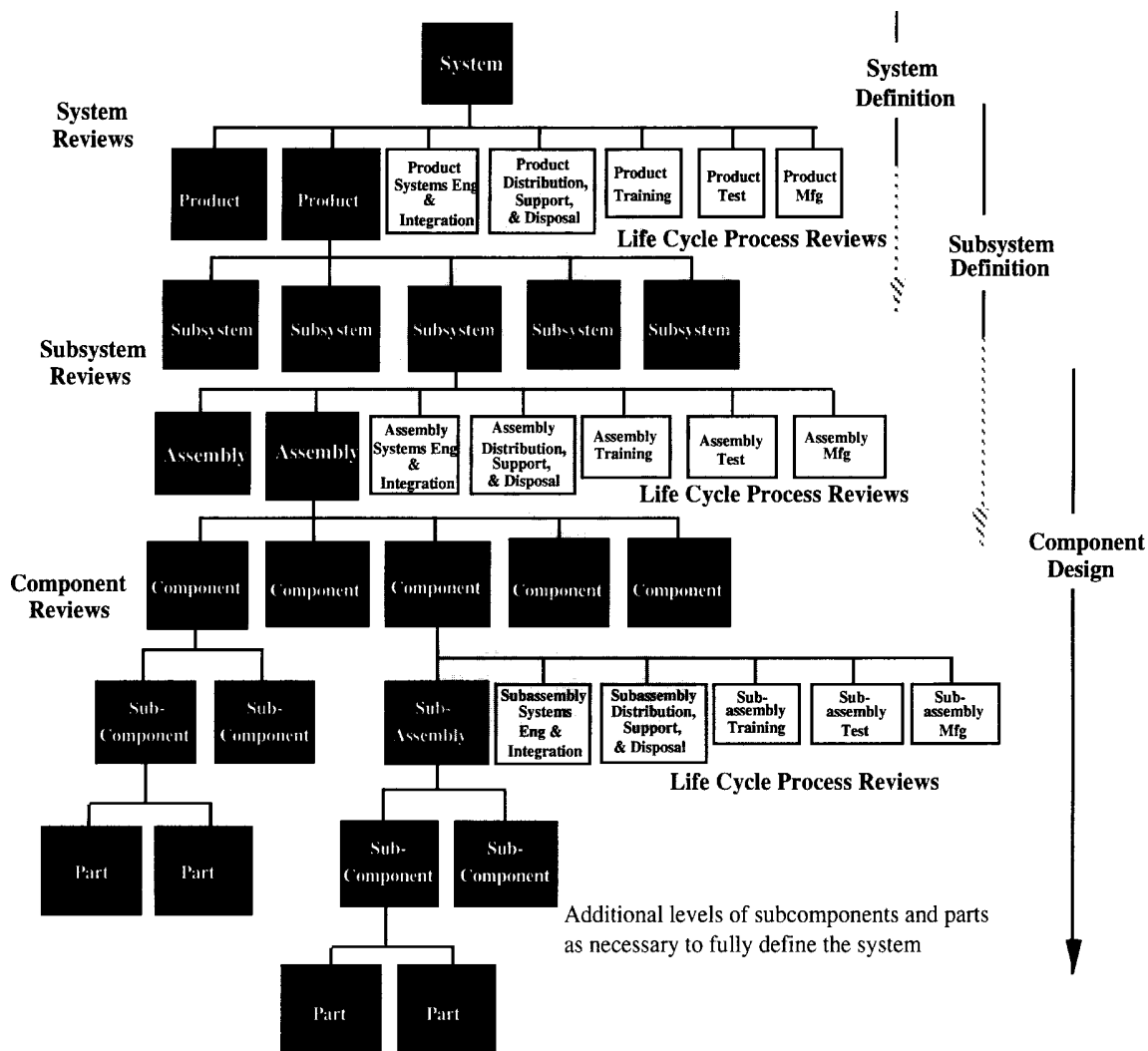


Figure 7—Typical system breakdown structure (SBS)