

IEEE Std 1220-1998

(Revision of
IEEE Std 1220-1994)

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

<p>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.</p>

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, *Chair*
Jerry Lake, *NCOSE Liaison*

Kenneth R. Ptack, *Editor*
Jim Lacy, *EIA Liaison*

George Friedman, *AIS Liaison*

Jim Armstrong

Bill Gess
Joel O'Rourke

John Snodderly

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

Stuart Arnold
Randy Case
Jim Cloud
Roger D. Larson

Jennifer McKneely
Amie Perry
Peter T. Poon

Arnold M. Ruskin
Jack Sivak
Ronald L. Wade
John Winters

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

Frank Akerman
Edward E. Barnett
H. Ronald Berlack
Richard E. Biehl
Mark Bilger
William J. Boll, Jr.
Ronald Braun
Bruce Brocka
Fletcher Buckley
Susan M. Burgess
Jaya R. Carl
Niva M. Carlson
Basilio Chen
Hu Cheng
John Chihonek
Tsun S. Chow
Lee Cooper
Taz Daughtrey
Paul I. Davis
Bostjan K. Derganc
Einar Dragstedt
Scott P. Duncan
Peter Eirich
Mokhtar S. El Assad
Richard Evans
John W. Fendrich
Jay Forster
Richard C. Fries
Simon Gabrihelidis
Yair Gershkovitch
Gregg Giesler
April Gillam
Shirley A. Gloss-Soler
John Garth Glynn
Rajeev Goel
Donald Gotterbarn

Michael P. Guba
David A. Gustafson
Jon D. Hagar
Ramzi Haraty
Manfred Hein
John W. Horch
David Johnson
Frank V. Jorgensen
Diana Kang
Richard Karcich
Ron S. Kenett
Judith S. Kerner
Scotty Kilbourne
F. L. King
Dwayne L. Knirk
Ed Kulas
John B. Lane
Boniface Lau
J. Dennis Lawrence
Michael Lines
William M. Lively
Ben Livson
James J. Longbucco
Joseph Maayan
Jukka Marjarvi
James W. McClean
Patrick D. McCray
Sue McGrath
Glen A. Meldrum
James E. Miller
Louis E. Miller
Celia H. Modell
Hironobu Nagano
Dennis E. Nickle
Mike Ottewill
Robert Parys
David E. Peercy

Alfred H. Peschel
Robert M. Poston
H. Reiche
Gregory Russel
Ronmald K. Saltgaver
Julio G. Sanz
David J. Schultz
Gregory D. Schumacher
Leonard W. Seagren
Carl S. Seddio
J. J. Selis
Keith Shewbridge
Robert W. Shillato
David M. Siefert
Mitchell W. Smith
Harry M. Sneed
Alfred R. Sorkowitz
Vijaya K. Srivastava
Richard Staunton
Jim Swanek
Douglas H. Thiele
Leonard L. Tripp
Margaret Updike
Ted J. Urbanowicz
Glenn D. Venables
Peter Voldner
L. Mark Walker
Delores Wallace
John W. Walz
Camille White-Partain
Paul R. Work
Forrest D. Wright
Natalie C. Yopconka
Wieder D. Yu
Janusz Zalewski
Peter F. Zoll

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.

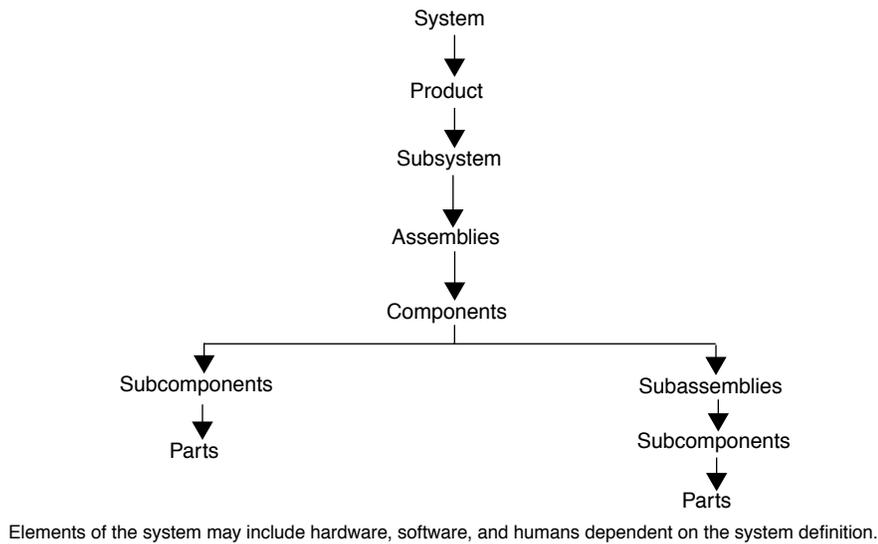


Figure 1—Hierarchy of elements within a system

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.

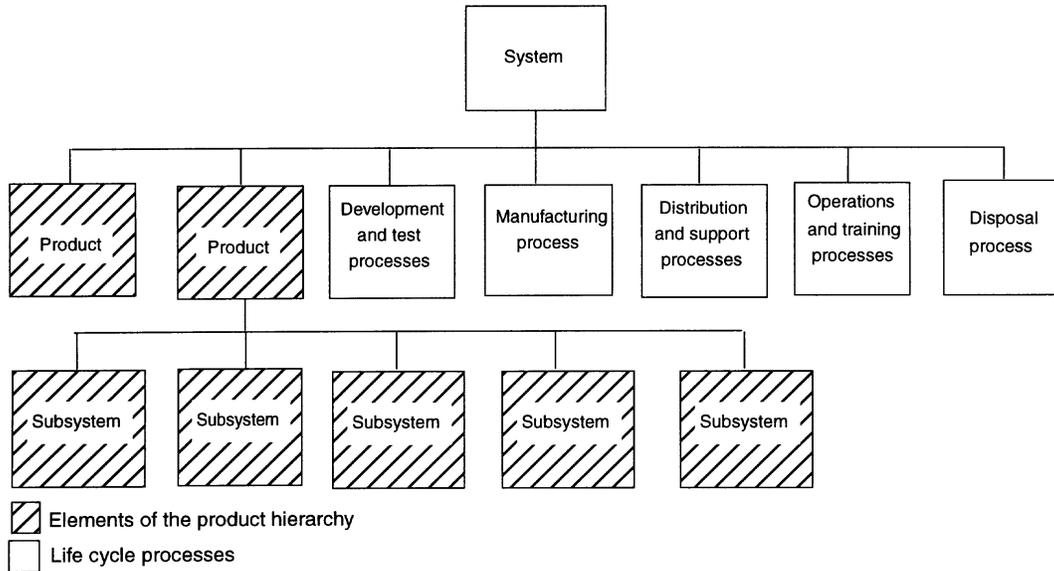


Figure 2—Basic building blocks of a system

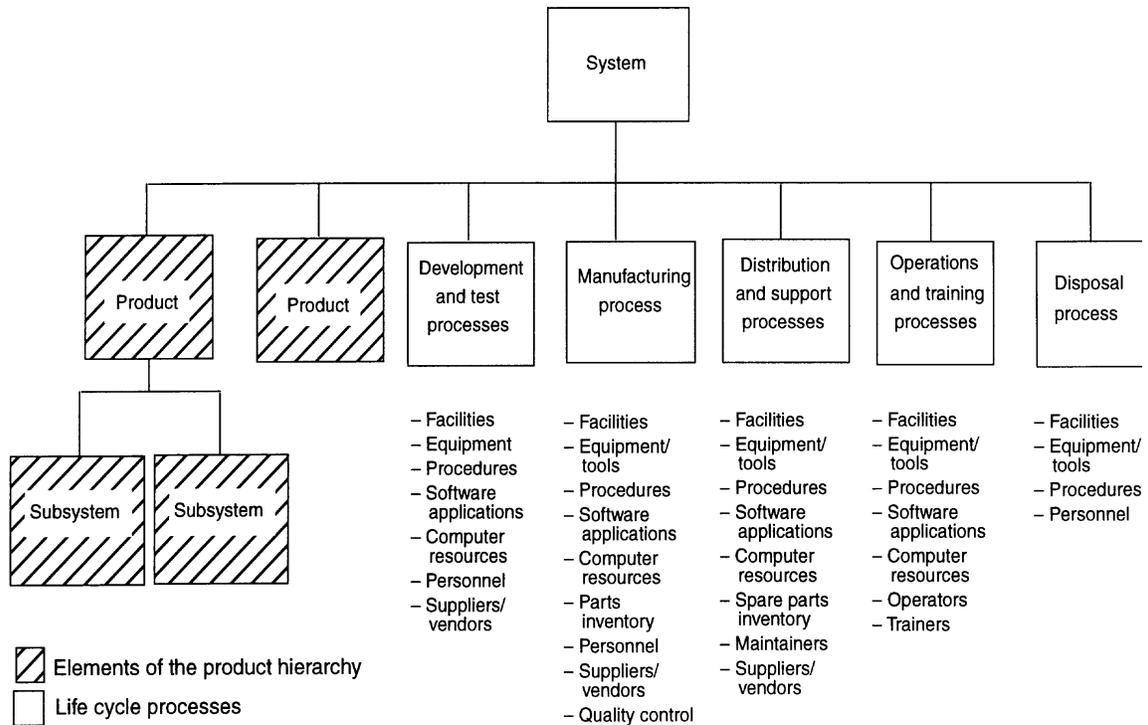


Figure 3—Life cycle process definition

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.

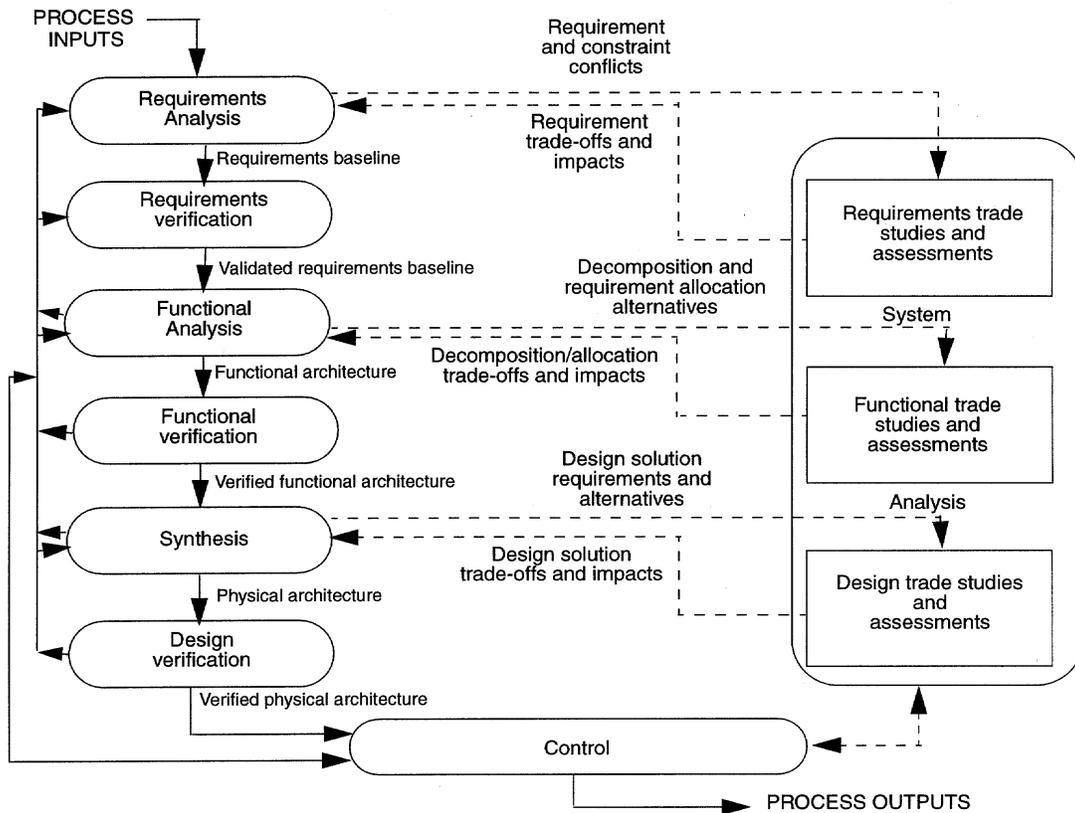


Figure 4—Systems engineering process (SEP)

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:
<p>A. Hardware</p> <ol style="list-style-type: none"> 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.
<p>B. Software</p> <ol style="list-style-type: none"> 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.
<p>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.</p> <ol style="list-style-type: none"> 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.
<p>D. Life cycle processes</p> <p>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.</p>

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.

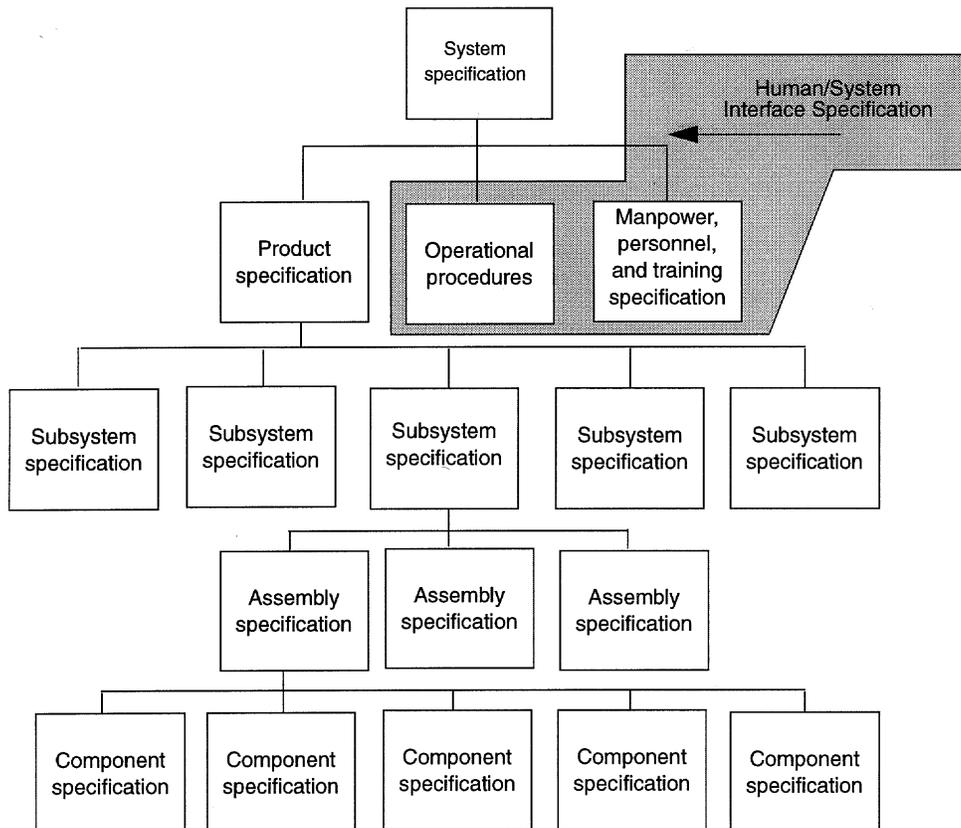


Figure 5—The specification tree

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.

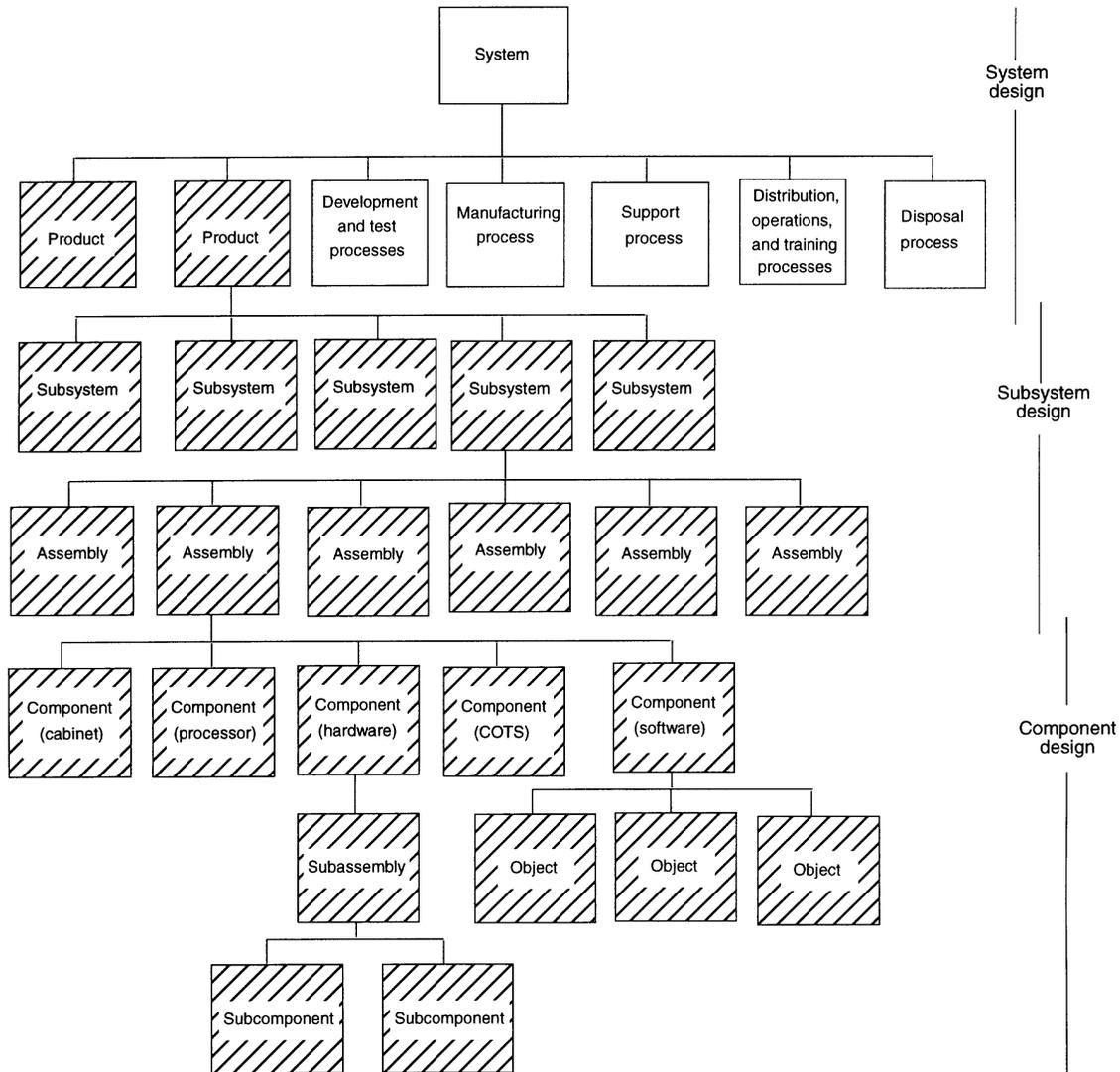


Figure 6—The system breakdown structure (SBS)

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.

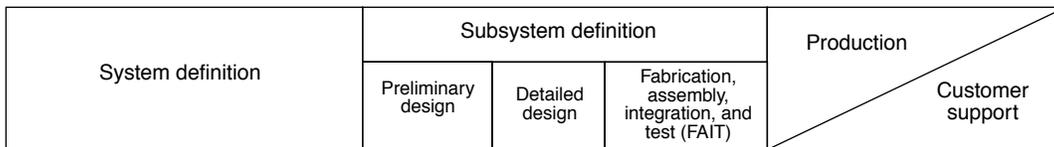


Figure 7— Typical system life cycle

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

Table 2—System definition

<p>Establish system definition</p> <ul style="list-style-type: none"> a) Select system concept b) Establish initial project and technical plans c) mitigate system risks d) Assess subsystem risks e) Identify subsystems and subsystem interfaces f) Identify human/system interface issues g) Define life cycle quality factors <ul style="list-style-type: none"> 1) Producibility 2) Verifiability 3) Ease of distribution 4) Usability 5) Supportability 6) Trainability 7) Disposability 8) Total ownership costs h) Revise engineering and technical plans for preliminary design
<p>Complete specifications</p> <ul style="list-style-type: none"> a) Complete system and product interface specifications b) Complete system and product specifications c) Complete subsystem interface specifications d) Complete preliminary subsystem specifications e) Complete preliminary human/system interface f) Complete preliminary manpower, personnel, and training
<p>Establish baselines</p> <ul style="list-style-type: none"> a) Establish system baseline b) Establish preliminary subsystem “design to” baselines
<p>Complete technical reviews</p> <ul style="list-style-type: none"> a) Complete alternative concept review b) Complete system definition review

enterprise should create and evaluate candidate alternative concepts that would satisfy a market opportunity or customer order. One or more of the alternative concepts should be selected for further system definition during this first stage of development. The risks associated with each alternative system should be assessed to include risk identification and quantification. Additionally, the enterprise should complete preliminary system and product specifications for each viable alternative, along with system- and product-interface specifications and a system baseline.

5.1.1.2 Initial engineering plans

The enterprise shall establish the necessary engineering and technical plans for the system life cycle, which include

- a) The activity accomplishment criteria for determining system definition progress assessment; and
- b) The allocation of technical resources among the systems engineering activities.

For precedented systems, the engineering plan, master schedule, and detail schedule are prepared for the full development life cycle, with the most detail for the current stage. For unprecedented systems, the engineering

plan, master schedule, and detail schedule are developed and evolved as the system is developed. Related technical plans should be developed, as appropriate, for manufacturing, logistics, computer resources, security, safety, reliability and maintainability, and training. Engineering and technical plans address development of the system products to satisfy the operational functions, as well as development of life cycle processes required to satisfy system-level development, manufacturing, test, distribution, support, training, and disposal functions.

5.1.1.3 Subsystems and subsystem interfaces

The enterprise shall identify the subsystems of each product and define the design and functional interface requirements among the subsystems and their corresponding performance requirements and design constraints. System product functional and performance requirements should be allocated among the subsystems so as to assure requirement traceability from the system products to their respective subsystems, and from subsystems to their parent product.

5.1.1.4 Identify human/system interface issues

The enterprise shall identify the interface requirements between humans and products or subsystems. These interface requirements include performance, workloads, design constraints, and usability.

5.1.1.5 System and subsystem risks

The enterprise shall mitigate system-level risks to include products risks that were assessed to be critical to system development during concept selection. For critical risks associated with products, the enterprise utilizes simulation, scale-model tests, or prototype tests to demonstrate mitigation of risks to an acceptable level. The enterprise should assess subsystem risks and prioritize critical risks based upon probability of occurrence and related consequences to cost, schedule, and/or performance.

5.1.1.6 Life cycle quality factors

The enterprise shall define the system life cycle quality factors that influence the system's capability to meet downstream requirements for producibility, test, ease of distribution (packaging, handling, transportation, storage, installation, and transition), usability, supportability, ease of training, and disposability. The system life cycle quality factors should be decomposed and allocated among the products, and then the subsystems, in a manner which ensures that quality factors traceability is maintained.

5.1.1.7 Revised project and engineering plans

The enterprise shall update necessary engineering and technical plans in response to changes based on SEP activities conducted during system definition, and to reflect planning for the next stage of development.

5.1.2 Specifications

The enterprise shall prepare and control the following specifications needed to guide the systems engineering efforts of the system definition stage.

5.1.2.1 System, product, and subsystem interface specifications

During the initial efforts of the concept/system definition stage, the enterprise shall complete, and place under change control, system and product interface specifications. The system interface specification should define the external functional and design interfaces for the system with respect to other systems. External interfaces unique to the selected system concept(s) shall be defined and documented in the system interface specification in a manner that recognizes the unique nature of these special interface elements. Product interface specifications should define the design and functional interface requirements between products of the system and life cycle sustainment equipment, such as special test equipment and training aids. The subsystem interface specifications should define design and functional interface requirements among subsystems for each product. The

enterprise should identify which interface requirements provide constraints to system/product/subsystem design, and which interfaces are sufficiently critical to be managed through interface control working groups.

5.1.2.2 System and product specifications

The enterprise should complete and control a system specification and a product specification for each product of the system. The system specifications should document the system requirements (functional and performance requirements, design characteristics, and design constraints) and the qualification requirements for each requirement in the system specification. The product specifications should document the allocation of system-level requirements to each product, and the qualification requirements for each requirement in a product specification. The qualification sections of each specification should identify the methods used to confirm that each system or product requirement has been satisfied under normal and abnormal conditions.

5.1.2.3 Preliminary subsystem specifications

The enterprise shall prepare a preliminary subsystem specification for each subsystem identified in the system design architecture. Preliminary subsystem specifications should identify subsystem requirements (functional and performance requirements, design characteristics, and design constraints) and the qualification requirements for each requirement in the specification. The qualification section of the specifications should identify the methods used to confirm that each subsystem requirement has been satisfied under normal and abnormal conditions.

5.1.2.4 Complete preliminary human/system interface specifications

The enterprise shall prepare a preliminary human/system interface specification for the interaction among humans and the hardware and software elements identified in the system design architecture. The qualification section of the specifications should identify the methods used to confirm that each human/system interaction requirement has been satisfied under normal and abnormal conditions. In addition, human/system interface specifications include interface requirements between humans and/or subsystems or components. These interfaces include human-human, human-hardware, and human-software. The requirements may include functional and performance requirement, and workload and design constraints.

5.1.2.5 Complete preliminary, manpower, personnel, and training specifications

The enterprise shall prepare a specification for the personnel required to operate, maintain, and support the system throughout its life cycle. Analysis shall determine if personnel with the appropriate knowledge, skills, and abilities are predicted to be available throughout the life cycle; the cognitive, physical, and sensory characteristics of the available personnel; the safety features required of the system; and the level of training required for such personnel.

5.1.3 Configuration baselines

The enterprise shall prepare and place under configuration control the following baselines needed to guide the technical efforts of the system definition stage.

5.1.3.1 System baseline

The enterprise shall establish, and place under control, the system baseline. This system-level configuration baseline should include the system interface specification, the product interface specifications, the system specification, the product specifications, and the integrated database that captures the design, data, models, metrics, changes, design rationale, and other pertinent information on decisions or clarification made to system requirements.

5.1.3.2 Subsystem design-to baselines

The enterprise shall generate a subsystem design-to baseline for each subsystem identified in the system architecture. Each design-to baseline should include applicable subsystem interface specifications, the related preliminary subsystem specification, and any subsystem drawings or sketches developed to define the system's products.

5.1.4 Technical reviews

The enterprise shall plan and conduct applicable technical reviews to assess the maturity of the development effort and to recommend whether the investment should be made to continue the development effort. The system-level reviews completed in this stage of development may be held in conjunction with a project management review.

5.1.4.1 Alternative concept review

An alternative concept review, if needed, should be completed by the enterprise to select a concept or concepts to which the system definition activities described in 5.1.1.2 through 5.1.3.2 are to be applied. During this review, each concept is evaluated based on

- a) Product and subsystem allocations being reasonable and providing a sound system concept;
- b) The capability of the concept to satisfy customer requirements and meet public expectations;
- c) The completion of the system and product interface specifications and preliminary system specification;
- d) The establishment of a system baseline;
- e) The assessed risks associated with the concept;
- f) The adequacy and completeness of systems analysis data to substantiate decisions made in defining the concept and establishing that the concept satisfies customer requirements and meet public expectations.

5.1.4.2 System definition review

A system definition review should be completed by the enterprise at the completion of the system definition stage for the purpose of determining whether the system definition is sufficiently mature to progress to subsystem definition. The system definition is reviewed to ensure that

- a) It is sufficiently mature to meet SEMS criteria;
- b) System-level risks have been adequately addressed to justify continued development;
- c) Trade-study data are adequate to substantiate that system requirements are achievable;
- d) Decisions made in arriving at the system definition configuration are well supported by analysis, test, and/or other technical data.

5.2 Preliminary design stage

The enterprise shall execute the preliminary design stage to initiate subsystem design and create subsystem-level specifications and design-to baselines to guide component development. The enterprise shall apply the SEP for the purpose of decomposing identified subsystem functions into lower-level functions and allocating functional and performance requirements to component-level functional and physical architectures in accordance with the following subclauses.

Each preliminary subsystem specification and preliminary design-to baseline should be evolved into a subsystem specification and a design-to baseline, respectively. Preliminary component specifications and build-to baselines should be defined for the components of the subsystem being developed. Final subsystem documents should include identification of recommended components and interfaces; resolution of sub-

system-level risks; assessment of component risks; and design for quality factors to include, as appropriate, producibility, verifiability, ease of distribution, usability, supportability, trainability, and disposability for each subsystem.

5.2.1 Preliminary subsystem definition

The enterprise shall apply the SEP, described in Clause 6, to each subsystem for the purpose of generating subsystem functional and physical architectures. Specific activities to be accomplished are listed in Table 3.

Table 3—Subsystem definition—preliminary design

Establish preliminary subsystem definitions <ol style="list-style-type: none"> a) Identify assemblies and assembly interfaces b) Identify components and component interfaces c) Mitigate subsystem risks d) Assess component risks e) Design for life cycle quality factors f) Complete preliminary drawings for each subsystem g) Identify human/systems interface issues h) Revise engineering and technical plans for detailed design
Complete specifications <ol style="list-style-type: none"> a) Update system and product specifications b) Complete subsystem and assembly specifications c) Complete component interface specifications d) Complete preliminary component specifications e) Update human/systems interface specifications f) Update manpower, personnel, and training specifications
Establish baselines <ol style="list-style-type: none"> a) Update system baselines b) Establish “design to” baselines c) Establish preliminary “build to” baselines
Complete technical reviews <ol style="list-style-type: none"> a) Complete subsystem preliminary design reviews b) Complete system preliminary design review

5.2.1.1 Assemblies and assembly interfaces

The enterprise shall identify the assemblies of each subsystem and define the design and functional interface requirements among the assemblies and their corresponding performance requirements and design constraints. Subsystem performance requirements are allocated among the assemblies so as to assure requirements traceability from subsystems to appropriate assemblies, and from assemblies to the parent subsystem.

5.2.1.2 Components and component interfaces

The enterprise shall identify the components of each assembly and define the design and functional interface requirements among the components and their corresponding performance requirements and design constraints. Assembly performance requirements are allocated among the components so as to assure requirement traceability from the assemblies to their respective components, and from components to their parent assembly.

5.2.1.3 Subsystem and component risks

The enterprise shall mitigate subsystem-level risks that were assessed to be critical to subsystem development during system definition, and shall assess and mitigate assembly risks associated with each subsystem. For critical subsystem/assembly risks, simulations, scale model tests, or prototype tests should be used to demonstrate mitigation to an acceptable risk level with respect to cost, schedule, and/or performance. The enterprise should assess component risks and prioritize critical risks based on the probability of occurrence and related consequences to cost, schedule, and/or performance.

5.2.1.4 Identify human/system interface issues

The enterprise shall identify the interface requirements between humans and assemblies or components. These interface requirements include performance, workloads, design constraints, and usability.

5.2.1.5 Life cycle quality factors

The enterprise shall identify and quantify subsystem life cycle quality factors that influence each subsystem's capability to meet downstream requirements for producibility, test, ease of distribution (packaging, handling, transportation, storage, installation, and transition), usability, supportability, ease of training, and disposability. The subsystem life cycle quality factors should be decomposed and allocated among the assemblies, and then the components, in a manner that ensures that quality-factor traceability is maintained.

5.2.1.6 Engineering and technical plans

The enterprise should update engineering and technical plans in order to respond to changes based on SEP activities conducted during subsystem definition, and to reflect planning for the next stage of development.

5.2.2 Subsystem specifications

The enterprise shall prepare and control the specifications needed to guide component design activities of the detailed design stage of subsystem definition.

5.2.2.1 System and product specifications

During the preliminary design stage, the enterprise should update and control all changes to established specifications. Typically for this stage in the life cycle, specifications under control include the system interface specification, subsystem interface specifications, product specifications, and the system specification.

5.2.2.2 Subsystem specifications

The enterprise should complete a subsystem specification for each subsystem, and place it under configuration control. These specifications should document the functional and performance requirements, design requirements or other imposed design constraints, and the qualification requirements for each subsystem. The qualification section of individual subsystem specifications should identify the methods used to confirm that each subsystem requirement has been satisfied under normal and abnormal conditions.

5.2.2.3 Component interface specifications

The enterprise should prepare, and place under configuration control, a component interface specification for each product component identified in the design architecture. Component interface specifications should document the functional and design interfaces among components, which shall be satisfied during component development. The enterprise should identify which assembly interface requirements provide constraints to component design and which shall have changes managed through interface control working groups.

5.2.2.4 Preliminary component specifications

The enterprise should prepare a preliminary component specification for each product component identified in the design architecture. The preliminary component specification should identify the functional and performance requirements, design requirements or constraints, and the qualification requirements for each performance requirement in a preliminary component specification. The qualification section of individual specifications should identify the methods that will be used to confirm that each component requirement has been satisfied under normal and abnormal conditions.

5.2.2.5 Update preliminary human/system interface specifications

The enterprise shall update human/system interface specifications for the interaction among humans and the hardware and software elements identified in the design architecture. The qualification section of the specifications should identify the methods used to confirm that each human/system interaction requirement has been satisfied under normal and abnormal conditions. In addition, human/system interface specifications include interface requirements between humans, subsystems, or components. These interfaces include human-human, human-hardware, and human-software. The requirements may include functional and performance requirement, and workload and design constraints.

5.2.2.6 Update preliminary, manpower, personnel, and training specifications

The enterprise shall update the specifications for the personnel required to operate, maintain, and support the system throughout its life cycle. Analysis shall determine if personnel with the appropriate knowledge, skills, and abilities are predicted to be available throughout the life cycle; the cognitive, physical, and sensory characteristics of the available personnel; the safety features required of the system; and the level of training required for such personnel.

5.2.3 Configuration baselines

The enterprise should place under configuration control the following baselines needed to guide the technical efforts of the subsystem definition stage.

5.2.3.1 System baseline

During the preliminary design stage, the enterprise should update and control all changes to the system baseline.

5.2.3.2 Design-to baseline

The enterprise evolves/refines and establishes the design-to baseline for each subsystem from the preliminary design-to baseline generated during system definition. This baseline includes the assembly and component interface specifications, subsystem specifications, assembly specifications, and the integrated database, which captures the design, data, models and tools used, metrics, changes, design rationale, and other pertinent information on decisions or clarification made to subsystem requirements.

5.2.3.3 Component build-to baseline

The enterprise evolves/refines a component build-to baseline for each component of a subsystem. This baseline includes the component interface drawings and the draft component specification.

5.2.4 Technical reviews

The enterprise should conduct applicable technical reviews to assess the maturity of the development effort and to determine whether the investment should be made to continue to detailed design. Technical reviews

below the system level should not normally be held as project management reviews, but held as strictly technical reviews.

5.2.4.1 Subsystem reviews

Subsystem reviews should be completed by the enterprise for each subsystem at the completion of its preliminary design stage. The purpose of each review is to assure that

- a) The subsystem definition is sufficiently mature to meet SEMS criteria;
- b) Component allocations and preliminary component specifications are reasonable and provide a sound subsystem concept;
- c) Subsystem risks have been assessed and mitigated to a level appropriate to continue development;
- d) Trade-study data are adequate to substantiate that subsystem requirements are achievable;
- e) Decisions made in arriving at the subsystem configuration definition are well supported by analysis and technical data.

5.2.4.2 System review

A system-level review should be completed by the enterprise after completion of subsystem reviews. This review is to determine whether the total system approach to detailed design satisfies the system baseline; unacceptable risks are mitigated; issues for all subsystems, products, and life cycle processes are resolved; and accomplishments and plans warrant continued development effort.

5.3 Detailed design stage

The enterprise shall execute the detailed design stage of the system life cycle to complete subsystem design down to the lowest component level, and create a component specification and build-to component baseline for each component. The outputs of this stage are used to guide fabrication of preproduction prototypes for development test. The enterprise shall apply the SEP, described in Clause 6, as many times as needed to decompose identified component functions into lower-level functions, and allocate functional and performance requirements throughout the resulting lower-level functional and design architectures.

Each preliminary component specification and build-to baseline generated during preliminary design of the subsystem should be evolved into a component specification and a build-to baseline, respectively. Final component documents should include identification of recommended parts and interfaces; resolution of component-level risks; and for each component, down to the lowest subcomponent, the design for quality factors to include, as appropriate, producibility, verifiability, ease of distribution, usability, supportability, trainability, and disposability.

5.3.1 Detailed subsystem definition

The enterprise shall apply the SEP to each component and its subcomponents for the purpose of generating component functional and design architectures. Specific activities to be accomplished are listed in Table 4.

5.3.1.1 Component definition

The enterprise should decompose the components of each assembly to a level sufficient for design completeness, complete the definition of each subcomponent and the component, and control the interfaces among the subcomponents. Component requirements should be allocated among the subcomponents in a manner which ensures that requirements traceability is maintained in both directions.

Table 4—Subsystem definition—detailed design

Establish detailed subsystem definitions <ul style="list-style-type: none"> a) Complete component definition (for hardware and software) b) Resolve component risks c) Design in life cycle quality factors d) Identify human/system interface issues e) Prepare integrated data package f) Revise engineering and technical plans for fabrication, assembly, integration, and test
Complete specifications <ul style="list-style-type: none"> a) Update system, product, subsystem, and assembly specifications b) Complete component specifications (for hardware and software) c) Update human/system interface specifications d) Update manpower, personnel, and training specifications
Establish baselines <ul style="list-style-type: none"> a) Update system and “design to” baselines b) Establish “build to” baselines
Complete technical reviews <ul style="list-style-type: none"> a) Complete component detailed design reviews b) Complete subsystem detailed design reviews c) Complete system detailed design review

5.3.1.2 Component risks

The enterprise should mitigate component-level risks that were assessed to be critical to component development during subsystem definition. For critical component risks, simulations, scale model tests, or prototype tests should be used to demonstrate mitigation to an acceptable risk level with respect to cost, schedule, and/or performance. The enterprise should assess and mitigate subcomponent risks and prioritize critical risks based on probability of occurrence and related consequences to cost, schedule, and/or performance.

5.3.1.3 Identify human/system interface issues

The enterprise shall identify the interface requirements between humans and components, subassemblies, or subcomponents, as necessary. These interface requirements include performance, workloads, design constraints, and usability.

5.3.1.4 Life cycle quality factors

The enterprise should identify and quantify component life cycle quality factors, which influence each component’s capability to meet downstream requirements for producibility, verifiability (test), ease of distribution (packaging, handling, transportation, storage, installation, and transition), usability, supportability, trainability, and disposability. The component life cycle quality factors should be decomposed and allocated among the component’s subcomponents, and then lower subcomponents, in a manner which ensures that quality-factors traceability is maintained.

5.3.1.5 Integrated data package

The enterprise should complete detailed documentation for each component and its subcomponents to satisfy functional architecture requirements as allocated to the component, component interface specifications, and the assembly specification. The integrated data package should be produced and captured in the integrated database, and should contain the detailed drawing, code listing, manual procedures, etc., as defined in 4.7.

5.3.1.6 Engineering and technical plans

The enterprise should update necessary engineering and technical plans in order to accommodate changes based on SEP activities conducted during detailed design for subsystem definition, and to reflect planning for the FAIT stage.

5.3.2 Specifications

The enterprise should prepare and control the following specifications needed to guide FAIT activities of subsystem definition.

5.3.2.1 System, product, subsystem, and assembly specifications

The enterprise should update and control all changes to approved specifications during the detailed design stage. Typically for this stage in the life cycle the approved specifications may include the system, subsystem, and component interface specifications; the system specification; and product, subsystem, and assembly specifications.

5.3.2.2 Component specifications

The enterprise should complete a component specification for each component identified in the design architecture. The component specification identifies the functional and performance requirements for the component, in addition to design requirements, or design constraints, and the qualification requirements for each performance requirement. The qualification section of individual specifications should identify the methods that will be used to confirm that each component requirement has been satisfied under normal and abnormal conditions.

5.3.2.3 Update human/system interface specifications

The enterprise shall update human/system interface specifications for the interaction among humans and the hardware and software elements identified in the design architecture. The qualification section of the specifications should identify the methods used to confirm that each human/system interaction requirement has been satisfied under normal and abnormal conditions. In addition, human/system interface specifications include interface requirements between humans, subsystems, or components. These interfaces include human-human, human-hardware, and human-software. The requirements may include functional and performance requirements, and workload and design constraints.

5.3.2.4 Update manpower, personnel, and training specifications

The enterprise shall update the specifications for the personnel required to operate, maintain, and support the system throughout its life cycle. Analysis shall determine if personnel with the appropriate knowledge, skills, and abilities are predicted to be available throughout the life cycle; the cognitive, physical and sensory characteristics of the available personnel; the safety features required of the system; and the level of training required for such personnel.

5.3.3 Configuration baselines

The enterprise should prepare, and place under configuration control, the following baselines needed to guide the technical efforts of the subsystem definition stage.

5.3.3.1 System and design-to baselines

The enterprise should update and control all changes to the established system baseline and to established design-to baselines.

5.3.3.2 Build-to baselines

The enterprise should establish the build-to baseline for each component from the build-to baseline generated during preliminary design. This baseline should include the component interface specifications, the component specifications, and the integrated database, which captures the design, data, models and tools used, metrics, changes, design rationale, and other pertinent information on decisions or clarification made to component requirements.

5.3.4 Technical reviews

The enterprise should conduct applicable technical reviews to assess the maturity of the development effort and to recommend whether the investment should be made to continue into FAIT. Technical reviews below the system level should normally not be held as project management reviews, but held as strictly technical reviews.

5.3.4.1 Component reviews

Component reviews should be completed by the enterprise for each component at the completion of the detailed design stage. The purpose of this review is to ensure that

- a) Each detailed component definition is sufficiently mature to meet measure of effectiveness/measure of performance (MOE/MOP) criteria;
- b) Component specifications are reasonable and provide a sound component concept;
- c) Component and related life cycle process risks have been assessed and mitigated to a level appropriate to support FAIT;
- d) Trade-study data are adequate to substantiate that detailed component requirements are achievable;
- e) Decisions made in arriving at the detailed component definition configuration are well supported by analysis and technical data.

5.3.4.2 Subsystem reviews

Subsystem-level reviews should be completed by the enterprise for each subsystem after completion of component reviews associated with the subsystem. This review is to determine whether the subsystem detailed design satisfies the design-to baseline; risks are mitigated and remaining risks are acceptable; issues for all components, assemblies, and life cycle processes are resolved; and accomplishments and plans warrant continuation with FAIT.

5.3.4.3 System review

A system-level review should be completed by the enterprise after completion of subsystem detailed design reviews. This review is to determine whether the detailed design of the system satisfies the system baseline; unacceptable risks are mitigated; issues for all subsystems, products, and life cycle processes are resolved; accomplishments and plans satisfy criteria for continuation of the technical effort; and the system is ready to continue into FAIT by having resolved outstanding product or life cycle process issues.

5.4 Fabrication, assembly, integration, and test (FAIT) stage

The enterprise shall apply the SEP, described in Clause 6, during this stage for the purpose of resolving product deficiencies when specifications for the system, product, subsystem, assembly, or component are not met, as determined by inspection, analysis, demonstration, or test. The purpose of the FAIT stage of subsystem definition is to verify that the products designed satisfy specifications. The major activities of this stage are shown in Table 5.

Table 5—Subsystem definition—fabrication, assembly, integration, and test (FAIT)

<p>Conduct system integration and test</p> <ul style="list-style-type: none"> a) Fabricate hardware components, implement software components b) Assemble, integrate, and test components and assemblies c) Assemble, integrate, and test subsystems and products d) Establish life cycle processes e) Analyze and fix failures/deficiencies and retest f) Update all specifications and baselines g) Revise engineering and technical plans for production
<p>Complete technical reviews</p> <ul style="list-style-type: none"> a) Complete component test readiness reviews b) Complete subsystem test readiness reviews c) Complete system test readiness review d) Complete component functional configuration audits e) Complete subsystem functional configuration audits f) Complete system functional configuration audit g) Complete component production approval reviews h) Complete subsystem production approval reviews i) Complete system production approval review

5.4.1 System integration and test

The enterprise shall execute integration activities to ensure that combining the lower-level elements results in a functioning and unified higher-level element, with logical and design interfaces satisfied. The enterprise shall execute the test activities to verify that the system meets system requirements. The enterprise completes this by first testing the components and then conducting tests at each level up to, and including, the total system. The enterprise shall progressively assemble and integrate subcomponents into complete components, components into assemblies, assemblies into subsystems, subsystems into products, and then, where meaningful, products and their life cycle processes and services into a complete system. At each level of assembly and integration, the components, assemblies, subsystems, products, and system should be subjected to sufficient testing to ensure operational effectiveness, usability, trainability, interface conformance, conformance with specified requirements, producibility, and supportability. The enterprise is responsible for the proper handling and disposal of test articles and all hazardous wastes generated by tests or used in conjunction with tests.

5.4.2 Analyze, fix, and retest

When a subcomponent, component, assembly, subsystem, or product fails to satisfy its requirements, the enterprise should analyze the deficiency to determine the cause of the problem and shall apply the SEP to resolve the problem. The enterprise should then retests the product to ensure operational effectiveness,

usability, trainability, interface conformance, conformance with specified requirements, producibility, and supportability.

5.4.3 Project and engineering plans

The enterprise should update necessary engineering and technical plans in order to respond to changes based on SEP activities conducted during the FAIT stage for subsystem definition, and to reflect planning for production.

5.4.4 Specifications

The enterprise should update and control all changes to approved specifications.

5.4.5 Configuration baselines

The enterprise should update and control all changes to the established baselines.

5.4.6 Technical reviews

The enterprise should plan and conduct applicable technical reviews to assess the maturity of the development effort, to determine readiness to conduct qualification testing, and to determine whether the investment should be made to continue into production. Technical reviews below the system level should not normally be held as project management reviews, but held as strictly technical reviews.

5.4.6.1 Test readiness reviews

Test readiness reviews (as needed for components, assemblies, subsystems, products, and the system) should be completed by the enterprise to assure that

- a) Test procedures comply with test plans and descriptions, demonstrate adequacy to accomplish test requirements, and satisfy specification qualification requirements;
- b) Pretest predictions and informal test results (if any) indicate testing shall confirm satisfaction of specification requirements;
- c) New or modified test support equipment, facilities, and procedural manuals, required to accomplish test and evaluation, should be available and should satisfy their requirements;
- d) Required specification, baseline, and other supporting documentation are complete and accurate.

5.4.6.2 Functional configuration audits

Functional configuration audits (as needed for components, subsystems, and the system) shall be completed by the enterprise to verify that products have achieved requirements; that they satisfy the characteristics as specified in specifications, interface specifications, and other baseline documentation; and that test plans and procedures were complied with.

5.4.6.3 Production approval reviews

A system level production approval review shall be completed by the enterprise after completion of product functional configuration audits to demonstrate that the total system (people, products, and processes) has been verified to satisfy specification and baseline requirements for each system level, and to confirm readiness for production, distribution, operations, support, training, continuing improvement (if applicable), and disposal. The enterprise should confirm that

- a) Issues for the components, assemblies, subsystems, products, and life cycle processes and services are resolved;

- b) Test procedures for components, assemblies, subsystems, and products were completed and were accurate;
- c) The system and products were confirmed ready for test;
- d) Tests were conducted in accordance with established procedures;
- e) An audit trail from design reviews, held after detailed design, is established with changes substantiated, and all component, subsystem, and system products meet specification requirements;
- f) Risk-handling procedures are satisfactory for production;
- g) Evolutionary development requirements and plans have been refined;
- h) Planning is complete and procedures, resources, and other requisite people, products, and processes are available (or programmed to be available) to initiate production, distribution, operations, support, training, disposal, and evolutionary development (if any).

5.5 Production and customer support stages

The enterprise shall apply the SEP, in accordance with Clause 6, to correct deficiencies discovered during production, assembly, integration, and acceptance testing of products and/or life cycle process products. The enterprise shall also apply the SEP during customer support to evolve the product to implement an incremental change, resolve product or service deficiencies, or to implement planned evolutionary growth. The major events of these two stages of a product's life cycle are shown in Figure 8.

Production	Customer support
Produce system products <ul style="list-style-type: none"> a) Perform production inventory and control activities b) Produce and assemble consumer products c) Correct product- and process-design deficiencies d) Dispose of by-products and wastes Complete technical reviews <ul style="list-style-type: none"> a) Complete component physical configuration audits b) Complete subsystem physical configuration audits c) Complete system physical configuration audits 	Provide customer support <ul style="list-style-type: none"> a) Provide customer service b) Provide "after market" products Complete system evolution <ul style="list-style-type: none"> a) Evolve design to <ul style="list-style-type: none"> 1) Make an incremental change 2) Resolve product deficiencies 3) Exceed competitive products

Figure 8—Production and customer support

5.5.1 System products

The enterprise should perform appropriate production inventory and control activities; production, assembly, integration, and acceptance test activities; and packaging, handling, storage, delivery, and installation activities to provide system products to consumers and support organizations. The enterprise should manage suppliers to ensure the timely delivery of products, materials, and services needed to carry out production activities.

5.5.1.1 Product and process deficiencies

The enterprise shall apply the SEP to correct product and/or process deficiencies found during production, acceptance testing, or distribution.

5.5.1.2 By-products and waste disposal

The enterprise should ensure the proper handling and disposal of hazardous wastes and materials generated by or used in production, acceptance testing, and distribution. In some cases, enterprises may be responsible for proper disposal of products after completing service life.

5.5.2 Technical reviews

Design configuration audits (as needed for components, subsystems, and the system) should be completed by the enterprise to assure the system elements conform to the technical documentation that defines the “build-to” baseline.

5.5.3 Customer support

Once products are in the field, the enterprise should continue to support customers with needed services and after-market products, and to maintain supplier relationships.

5.5.4 System evolution

The enterprise shall apply the SEP during customer support to make incremental improvements to fielded products and services; to resolve product or process deficiencies discovered during consumer use of products or during service activities; and to make changes to products and services to compete with products and services offered by competitors, and to ensure personnel with the appropriate knowledge, skills, and abilities to support the evolving product are available.

5.5.4.1 Revise project and engineering plans

The enterprise should update necessary engineering and technical plans in order to respond to changes based on SEP activities conducted during production and customer support.

5.5.4.2 Specifications

The enterprise should update and control all changes to approved specifications.

5.5.4.3 Configuration baselines

The enterprise should update and control all changes to established baselines.

5.6 Simultaneous engineering of life cycle processes

The enterprise shall accomplish planning activities and apply the SEP, in accordance with Clause 6, to develop life cycle processes and services for system product development, production, test, distribution, support, training, and disposal. Life cycle processes and services include such items as special tooling and equipment for manufacturing or maintenance; special processes for manufacturing; software and hardware for support equipment or training or test simulators; training or maintenance manuals; development, manufacturing, and test plans; facilities for test, manufacturing, or disposal; and procedures for services related to each downstream activity. The life cycle processes and services are needed to enable the capability of products to be fully realized throughout their life cycle. The development of life cycle processes should be delayed by the enterprise until requirements are defined for the products the life cycle processes support. The phasing for the simultaneous engineering of life cycle process products and services is presented in Figure 9.

Although development of life cycle process definitions may not be initiated prior to when the product (system product, subsystem assembly, or component subassembly) for which it is intended to support is defined, the enterprise shall schedule applicable downstream life cycle processes to be available at the time needed to

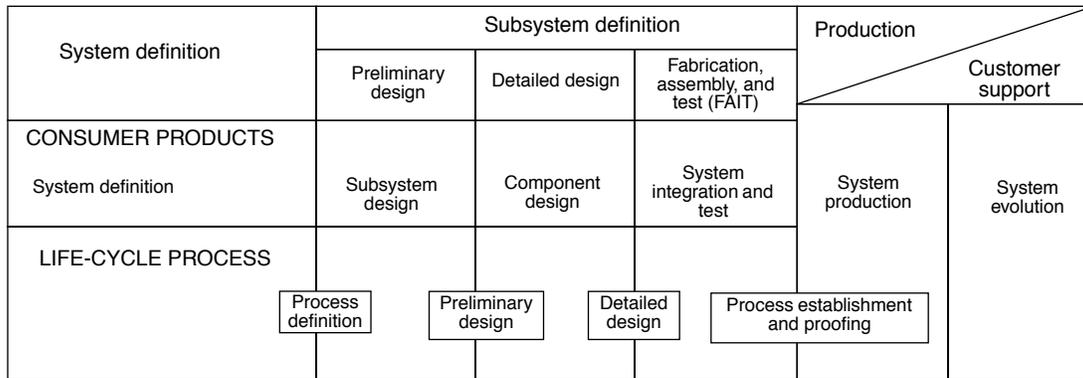


Figure 9—Simultaneous engineering of life cycle processes

enable the process to be supportive of the product for which it is intended. Since most life cycle processes are not as complex as the product for which they are intended to support, the development cycle should be shorter and should be available when needed.

5.6.1 Life cycle process development

The enterprise should initiate development of applicable downstream life cycle processes for development, production, test, distribution, support, training, and disposal to provide life cycle support to products and their subsystems, and for assemblies and their components. If system elements are being procured from suppliers or subcontractors, then consideration for life cycle support of the system elements should be addressed. Each life cycle process goes through the same development events and activities as described for products in 5.1 through 5.5, including technical reviews.

5.6.2 Specifications

The enterprise should prepare and control specifications of the same type, as described in 5.1 through 5.5, for the life cycle processes and services.

5.6.3 Baselines

The enterprise should prepare and control baselines of the same type, as described in 5.1 through 5.5, for the life cycle processes and services.

6. The systems engineering process (SEP)

This clause describes the detailed requirements of the SEP (see Figure 4). For each subprocess of the SEP, a figure is provided to diagram the general flow of tasks for the subprocess. The enterprise should tailor the activities of each task by adding or deleting activities, or tailor the subprocess tasks by adding or deleting tasks, according to the scope of the project.

6.1 Requirements analysis

The enterprise shall perform requirements analysis for the purpose of establishing what the system shall be capable of accomplishing; how well system products should perform in quantitative, measurable terms; the environments in which system products operate; the human/system interfaces requirements; the physical/aesthetic characteristics; and constraints that affect design solutions. The market needs, requirements, and constraints are derived from customer expectations, project and enterprise constraints, external constraints, and higher-level system requirements. These are documented in a requirements baseline. The requirements baseline guides the remaining activities of the SEP and represents the definition of the problem that should

be solved. The tasks associated with requirements analysis are identified in Figure 10. The enterprise assesses and analyzes inputs defined in tasks 6.1.1 through 6.1.9 to identify cost, schedule, and performance risks; to define functional and performance requirements; and identify conflicts. Trade-off analyses are conducted to resolve such conflicts so as to arrive at a balanced requirements baseline. The trade-off analysis and risk assessment, analysis, and handling tasks are discussed in 6.7. For each application of the SEP, the enterprise refines previously defined requirements for upper levels of the system architecture, as appropriate, and defines requirements for the system under development (refer to the discussion in 1.3).

6.1.1 Define customer expectations

The enterprise defines and quantifies the customer expectations for the system. Customer expectations may come from marketing, a customer order, a recognized market opportunity, direct communications from customers, or the requirements from a higher-level system. Customer expectations include

- a) What the customer wants the system (product(s), life cycle processes, and desired quality factors) to accomplish (functional requirements);
- b) How well each function should be accomplished (performance requirements);
- c) The natural and induced environments in which the product(s) of the system operates or may be used;
- d) Constraints (e.g., funding; cost or price objectives; schedule; technology; nondevelopmental and reusable items; design characteristics; hours of operation per day; on-off sequences; external interfaces; and specified existing equipment, procedures, or facilities related to life cycle processes).

The definition of customer expectations should be balanced with an analysis of their effects on the overall system design and performance, as well as on the human engineering; knowledge, skills, and abilities; availability; reliability; safety; and training requirements of the humans required to support life cycle processes.

6.1.2 Define project and enterprise constraints

The enterprise identifies and defines project and enterprise constraints that impact design solutions.

6.1.2.1 Project constraints

Project constraints may include

- a) Approved specifications and baselines developed from prior applications of the SEP;
- b) Updated engineering and technical plans;
- c) Team assignments and structure;
- d) Automated tools availability or approval for use;
- e) Control mechanisms;
- f) Required metrics for measuring technical progress.

6.1.2.2 Enterprise constraints

Enterprise constraints may include

- a) Management decisions from a preceding technical review;
- b) Enterprise general specifications, standards, or guidelines;
- c) Policies and procedures;
- d) Domain technologies;
- e) Established life cycle process capabilities;
- f) Physical, financial, and human resource allocations to the technical effort.

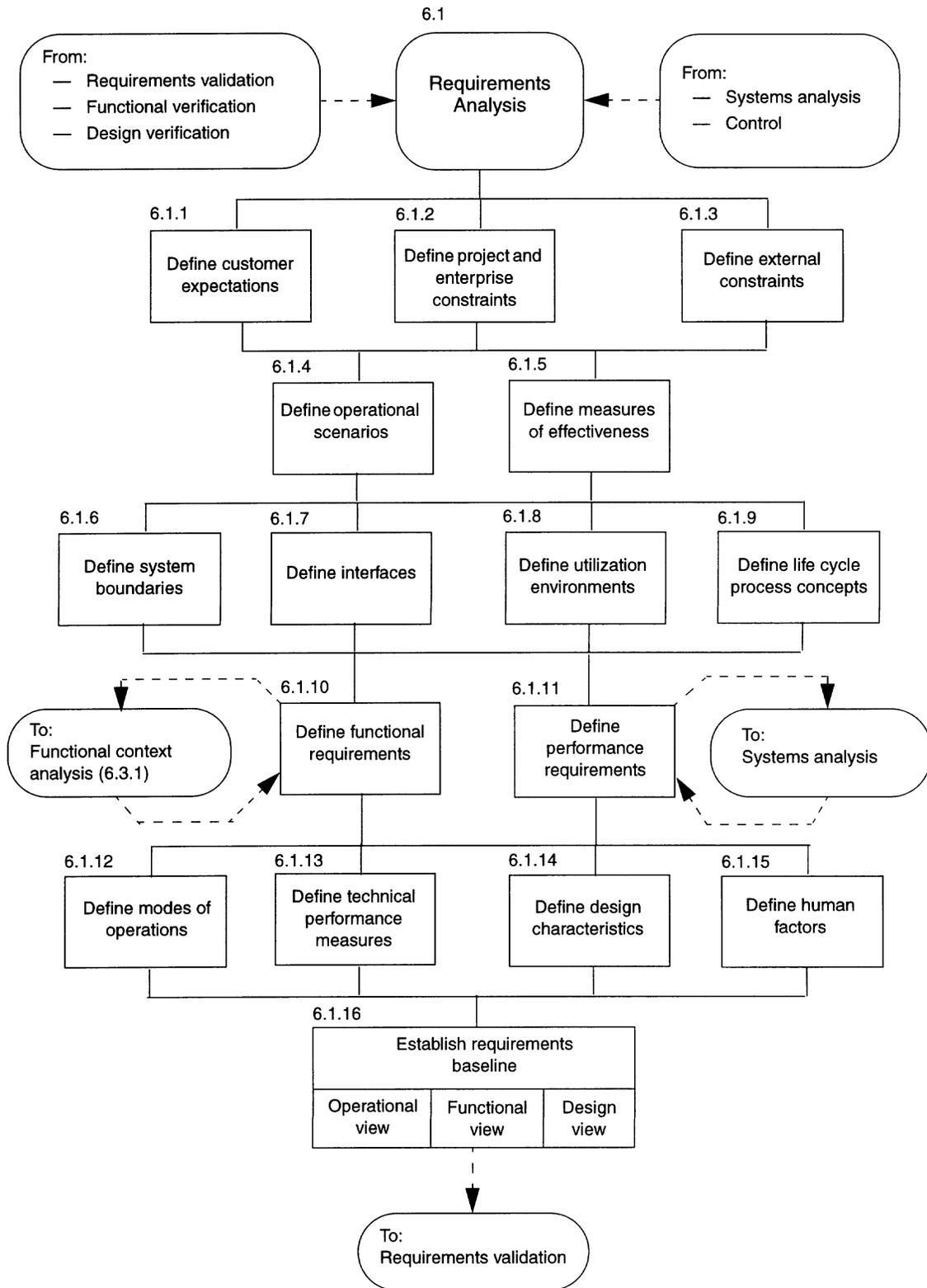


Figure 10—Requirements analysis process

The enterprise may also impose long-range planning constraints on the technical effort, which may demand an evolutionary development strategy in order to achieve project or enterprise goals.

6.1.3 Define external constraints

The enterprise identifies and defines external constraints that impact design solutions or the implementation of SEP activities. These constraints include

- a) Public and international laws and regulations;
- b) The technology base;
- c) Industry, international, and other general specifications, standards, and guidelines;
- d) Human-related specifications, standards, and guidelines;
- e) Human availability, recruitment, and selection;
- f) Competitor product capabilities.

6.1.4 Define operational scenarios

The enterprise identifies and defines the operational scenarios that define the range of the anticipated uses of system product(s). For each operational scenario, the enterprise defines expected interactions with the environment and other systems; human tasks and task sequences; and physical interconnections with interfacing systems, platforms, or products.

6.1.5 Define measures of effectiveness (MOEs)

The enterprise defines system effectiveness measures that reflect overall customer expectations and satisfaction. Key MOEs may include performance, safety, operability, usability, reliability, maintainability, time and cost to train, workload, human performance requirements, or other factors.

6.1.6 Define system boundaries

The enterprise defines

- a) Which system elements are under design control of the enterprise and which fall outside their control; and
- b) The expected interactions among system elements under design control and external and/or higher-level and interacting systems outside the system boundary.

6.1.7 Define interfaces

The enterprise defines the functional and design interfaces to external and/or higher-level and interacting systems, platforms, humans, and/or products in quantitative terms. Mechanical, electrical, thermal, data, communication-procedural, human-machine, and other interactions are included.

6.1.8 Define utilization environments

The enterprise defines the utilization environments for each of the operational scenarios. All environmental factors, natural or induced, which may affect system performance, should be identified and defined. Factors, which ensure that the system minimizes the potential for human or machine errors or failures that cause injurious accidents or death, and impart minimal risk of death, injury, or acute chronic illness, disability, and/or reduced job performance of the humans who support the system life cycle, are identified. Specifically, weather conditions (e.g., rain, snow, sun, wind, ice, dust, and fog), temperature ranges, topologies (e.g., ocean, mountains, deserts, plains, and vegetation), biological (e.g., animal, insects, birds, and fungi), time (e.g., day, night, and dusk), induced (e.g., vibration, electromagnetic, acoustic, and chemical), or other envi-

ronmental factors are defined for possible locations and conditions where the system may be operated. Effects on hardware, software, and humans should be assessed for impact on system performance and life cycle processes.

6.1.9 Define life cycle process concepts

The enterprise analyzes the outputs of tasks 6.1.1 through 6.1.8 to define life cycle process requirements necessary to develop, produce, test, distribute, operate, support, train, and dispose of system products under development.

6.1.9.1 Manpower

The enterprise identifies and defines the required job tasks and associated workload used to determine the number and mix of humans who support the system life cycle processes.

6.1.9.2 Personnel

The enterprise evaluates and determines the human experiences, aptitudes, knowledge, skills, and abilities required to perform the job tasks that are associated with the humans who support the system life cycle processes.

6.1.9.3 Training

The enterprise identifies and develops the instruction, education, and on-the-job or team training necessary to provide humans and teams with knowledge and job skills needed to support the system life cycle processes at the specified levels of performance. This includes the tools, devices (including embedded training systems), training simulators, techniques, procedures, and training materials and technical manuals to be developed and employed to provide training for all required tasks.

6.1.9.4 Human engineering

The enterprise identifies and accounts for those human cognitive, physical, and sensory characteristics of the humans who support the system life cycle that directly contribute to, or constrain, system performance and that impact human-machine interfaces.

6.1.9.5 Safety

The enterprise accounts for the system design features that create significant risks of death, injury, or acute chronic illness, disability, and/or reduce job performance of personnel who operate, maintain, or support the system.

6.1.10 Define functional requirements

The enterprise performs functional context analysis (6.3.1) for the purpose of defining what the system should be able to do (the functional requirements). The functions identified are used in 6.1.11 to define how well the functions must be performed and to establish the performance requirements. The functions identified through functional context analysis shall be further decomposed during functional decomposition (6.3.2) to provide a basis for identifying and assessing design alternatives. All requirements of a system shall involve a functional and performance aspect, and it is viewing system requirements as having both functional and performance aspects which ensures that the requirements are complete, consistent, and verifiable.

6.1.11 Define performance requirements

The enterprise defines the performance requirements for each function of the system. Performance requirements describe how well functional requirements must be performed to satisfy the MOEs. These performance

requirements are the MOPs that are allocated to subfunctions during functional decomposition analysis and that are the criteria against which design solutions [derived from synthesis (see 6.5)] are measured. There are typically several MOPs for each MOE, which bind the acceptable performance envelope.

6.1.12 Define modes of operation

The enterprise defines the various modes of operation (embedded training capability, fully operational, etc.) for the system products under development. The conditions (environmental, configuration, operational, etc.), which determine the modes of operation, are defined.

6.1.13 Define technical performance measures (TPMs)

The enterprise identifies the TPMs, which are key indicators of system performance. Selection of TPMs should be limited to critical MOPs that, if not met, put the project at cost, schedule, or performance risk. Specific TPM activities are integrated into the SEMS to periodically determine achievement to date and to measure progress against a planned value profile.

6.1.14 Define design characteristics

The enterprise identifies and defines required design characteristics (e.g., color, texture, size, anthropomorphic limitations, weight, and buoyancy) for the system products under development. The enterprise identifies which design characteristics are constraints and which can be changed based on trade-off analyses.

6.1.15 Define human factors

The enterprise identifies and defines human-factor considerations (e.g., design space limits, climatic limits, eye movement, reach, ergonomics, cognitive limits, and usability) that affect operation of the system under development. The enterprise identifies which human factors are constraints and which can be changed based on trade-off analyses.

6.1.16 Establish requirements baseline

The output of tasks 6.1.1 through 6.1.15 is recorded in three views (operational, functional, and design) to form a requirements baseline that establishes the system problem to be solved by the enterprise. The operational view describes how the system products serve their users. It establishes who operates and supports the system and its life cycle processes, and how well and under what conditions the system products are to be used. The functional view describes what the system products do to produce the desired behavior described in the operational view and provides a description of the methodology used to develop the view and decision rationale. The design view describes the design considerations of the system products development and establishes requirements for technologies and for design interfaces among equipment and among humans and equipment. The content of these views may include the following:

- a) Operational view
 - 1) Operational need description;
 - 2) Results of system operational analyses;
 - 3) Operational sequences/scenarios (best portrayed in pictures), which include utilization environments, MOEs, and how the system products should be used;
 - 4) Conditions/events to which system products should respond;
 - 5) Operational constraints, including MOEs;
 - 6) Identified human roles, including job tasks and skill requirements;
 - 7) Training requirements, including how humans will be trained to be a part of the system and support system life cycle processes through formal, informal, embedded, on-the-job, or other forms of training;
 - 8) Identification of what operations are required to ensure safety;

- 9) Life cycle process concepts to include MOEs, critical MOPs, and already existing products and services;
 - 10) Operational interfaces with other systems, platforms, humans, and/or products;
 - 11) System boundaries.
- b) Functional view
- 1) Functional requirements that describe what system products and life cycle processes must do or accomplish;
 - 2) Performance requirements including qualitative (how well), quantitative (how much, capacity), and timelines or periodicity (how long, how often) requirements;
 - 3) Functional sequences for accomplishing system objectives;
 - 4) TPM criteria;
 - 5) Functional interface requirements with external, higher-level, or interacting systems, platforms, humans, and/or products;
 - 6) Modes of operations;
 - 7) Functional capabilities for planned evolutionary growth.
- c) Design view
- 1) Previously approved specifications and baselines;
 - 2) Design interfaces with other systems, platforms, humans, and/or products;
 - 3) Human system engineering elements, including safety, training, and knowledge, skills, and abilities required to accomplish functions of the system, and characteristics of information displays and operator controls;
 - 4) Characterization of operator(s) and support personnel including special design requirements and applicable movement, or visual or workload limitations;
 - 5) Characterization of information displays and operator controls;
 - 6) System characteristics including design limitations (capacity, power, size, weight); technology limitations (precision, data rates, frequency, language); inherent human limitations (physical and cognitive workload, perceptual abilities, and reach and anthropometric limitations); and standardized end items, nondevelopmental items, and reusability requirements;
 - 7) Design constraints, including project, enterprise, and external constraints, which limit design solutions and/or development procedures;
 - 8) Design capabilities and capacities for planned evolutionary growth.

6.2 Requirements validation

The enterprise shall perform a validation of the requirements baseline established during requirements analysis. Validation consists of the following two types of activities:

- a) Evaluation of the requirements baseline to ensure that it represents identified customer expectations and project, enterprise, and external constraints; and
- b) Assessment of the requirements baseline to determine whether the full spectrum of possible system operations and system life cycle support concepts has been adequately addressed.

When voids in needs, constraints, etc. are identified or needs are not properly addressed, requirements analysis and validation are repeated until a valid requirements baseline is generated. The validated requirements baseline is documented in the integrated database and is an input to functional analysis. The tasks associated with validation are identified in Figure 11.

6.2.1 Compare to customer expectations

The enterprise analyzes and compares the established requirements baseline against customer expectations to ensure that the technical requirements adequately represent the customer's needs, requirements, and constraints for system products and life cycle support concepts. This involves direct customer involvement (end-user, marketing, etc.) and/or customer-provided requirement documents.

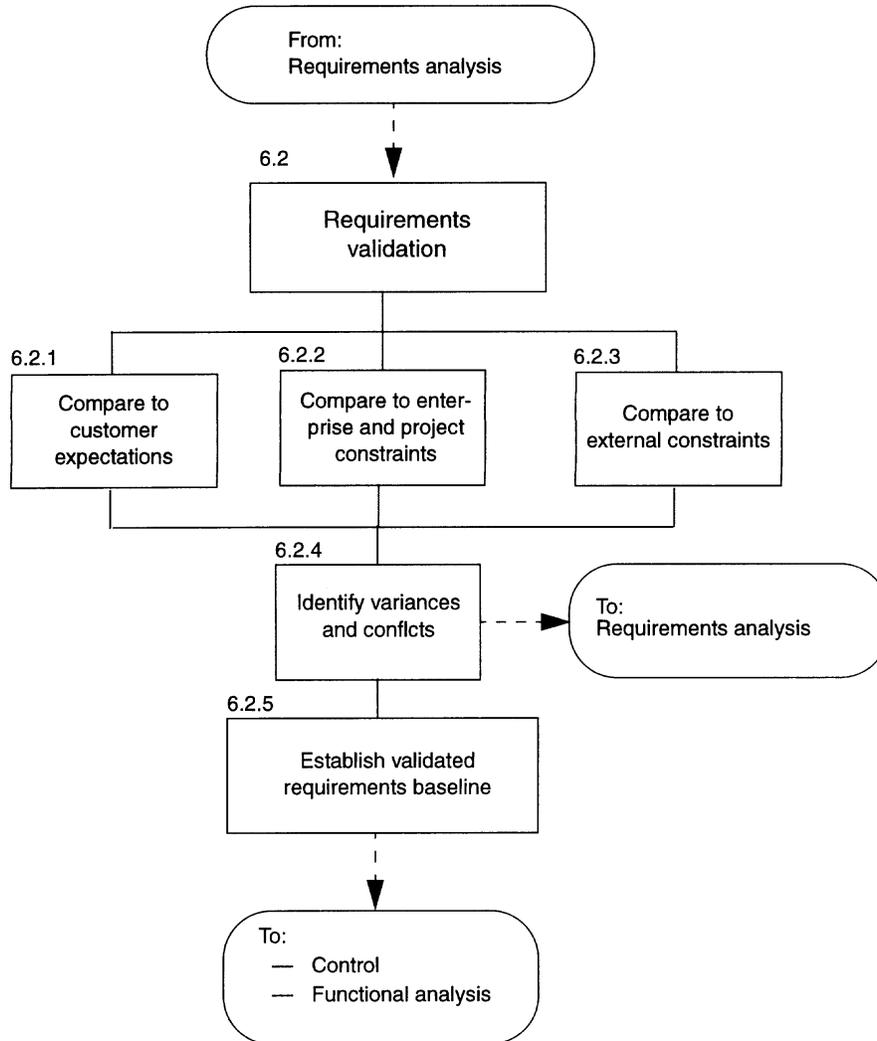


Figure 11—Requirements validation

6.2.2 Compare to enterprise and project constraints

The enterprise analyzes and compares the established requirements baseline against enterprise and project constraints to ensure that the technical requirements correctly represent, and stay within, enterprise and project policies and procedures, acceptable risk levels, plans, resources, technology limitations, objectives, decisions, standards, or other documented constraints.

6.2.3 Compare to external constraints

The enterprise analyzes and compares the established requirements baseline against external constraints to ensure that the specified technical requirements correctly represent, and stay within, applicable national and international laws (including environmental protection, hazardous material exclusion lists, waste handling, and social responsibility laws); correctly state external interface requirements with existing or evolving systems, platforms, or products; include applicable general specification and standard provisions affecting the development; and adequately define competitive product capabilities and characteristics.

6.2.4 Identify variances and conflicts

The enterprise identifies and defines variances and conflicts that arise out of the validation tasks 6.2.1 through 6.2.3. Each variance or conflict should be resolved by iterating through requirements analysis to refine the requirements baseline.

6.2.5 Establish validated requirements baseline

Once the established requirements baseline variations and conflicts are satisfactorily resolved, the requirements baseline is considered valid. This validated requirements baseline is then used as input to functional analysis (6.3) and documented in the integrated database.

6.3 Functional analysis

The enterprise shall perform functional analysis to accomplish the following two related objectives:

- a) To describe the problem defined by requirements analysis in clearer detail; and
- b) To decompose the system functions to lower-level functions that shall be satisfied by elements of the system design (e.g., subsystems, components, or parts).

This is accomplished by translating the validated requirements baseline into a functional architecture. The functional architecture describes the functional arrangements and sequencing of subfunctions resulting from decomposing (breaking down) the set of system functions to their subfunctions. Functional analysis should be performed without consideration for a design solution. Groups of subfunctions generated during synthesis (6.5) set the criteria that guide the definition of product and subsystem solutions. The tasks of functional analysis are identified in Figure 12.

6.3.1 Functional context analysis

The enterprise analyzes each system function to determine the responses (output) of the system to stimuli (inputs) necessary to accomplish system objectives.

6.3.1.1 Analyze functional behaviors

Analyses are conducted to understand the functional behavior of the system under various conditions and to assess the integrity of the functional architecture. Analyses should involve the simulation or stimulation of functional models utilizing operational scenarios that expose the models to a variety of stressful and non-stressful situations that reflect anticipated operational usage and environments.

6.3.1.2 Define functional interfaces

As system functions are decomposed into functions, interfaces between interacting functions are created. The enterprise identifies these interfaces and defines their functional interactions, such as start and end states or inputs and outputs.

6.3.1.3 Allocate performance requirements

Performance requirements are divided into allocable sets and are directly allocated to functions. Requirements that are not directly allocable, such as range, should be translated into derived performance requirements, such as fuel capacity, engine efficiency, and vehicle resistance, through appropriate engineering techniques and analyses. The enterprise documents the allocation of system performance requirements to functions to provide traceability and to facilitate later changes.

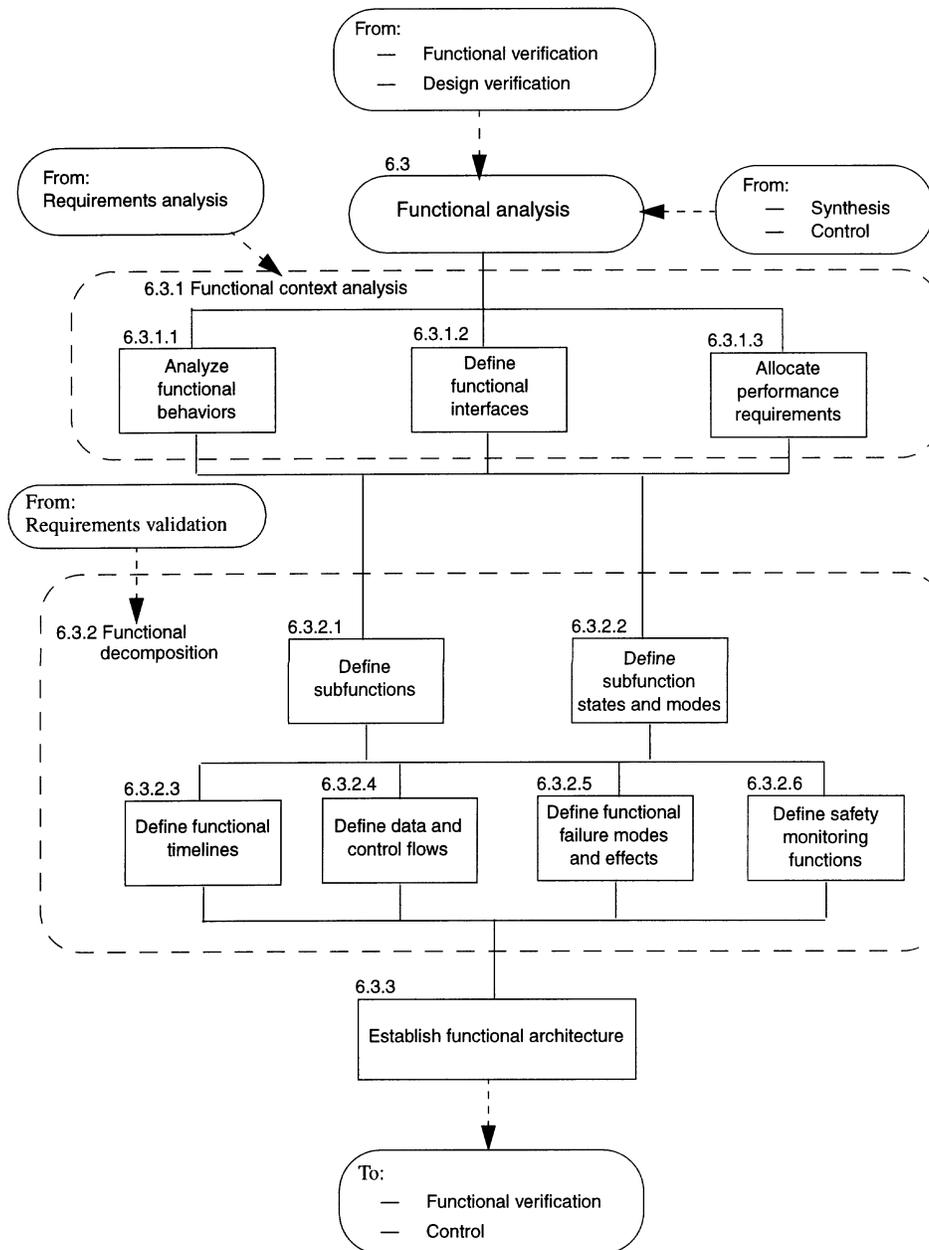


Figure 12—Functional analysis process

6.3.2 Functional decomposition

The enterprise decomposes the system into subfunctions with the intent of defining

- a) Alternative subfunction arrangements and sequences;
- b) Their functional interfaces;
- c) Their performance requirements.

The extent of functional decomposition depends on establishing a clear understanding of what the system must accomplish. Trade-off analyses and risk analyses (6.7) are performed to select a balanced set of subfunctions and to allocate performance requirements to subfunctions to assure that the functional architecture satisfies the system requirements.

6.3.2.1 Define subfunctions

Functions are decomposed in terms of their functional behaviors, states and modes of operation, functional timelines, conditions for control of data flow, functional failure modes and effects, and potential hazard monitoring functions that are needed. Alternative arrangements and sequencing of subfunctions are explored to assure a balanced set of subfunctions. The enterprise analyzes resulting subfunction arrangements to determine the degree of redundancy. Identified redundant functions, not specifically needed for safety, reliability, or other critical requirement, should be eliminated. The enterprise selects the best functional architecture and documents it in the integrated database.

6.3.2.2 Define subfunction states and modes

The enterprise analyzes the functional architecture to identify and define the states and modes for which subfunctions exhibit different behaviors. The analyses include state or mode transitions between start and end conditions of a subfunction, or aggregate of subfunctions.

6.3.2.3 Define functional timeline

The enterprise analyzes sequences of subfunctions and their behaviors to identify and define a functional timeline for each operational scenario. The ranges for the execution time for each subfunction and the conditions that cause normal and abnormal performance are identified in support of the functional timeline.

6.3.2.4 Define data and control flows

The enterprise analyzes sequences of subfunctions and their behaviors to identify and define data flows among the subfunctions for each operational scenario. These data flows are captured in a data-flow diagram or related object-oriented notation. The execution control of the functional architecture is identified and defined for each operational scenario and captured in a control-flow diagram or related object-oriented notation.

6.3.2.5 Define functional failure modes and effects

The enterprise analyzes and prioritizes potential functional failure modes to define failure effects and identify the need for fault detection and recovery functions. Functional reliability models are established to support the analysis of system effectiveness for each operational scenario. Failures, which represent significant safety, performance, or environmental hazards, are modeled to completely understand system impacts.

6.3.2.6 Define safety-monitoring functions

The enterprise analyzes subfunctions and aggregates of subfunctions to identify operational hazards that could result in personal injury, property or product damage, or environmental impacts. Additional functional requirements are derived and defined for monitoring dangerous operational conditions, or notifying or warning operators of impending hazards.

6.3.3 Establish functional architecture

The enterprise establishes the functional architecture, appropriate to the level of development, to define the allocation of performance requirements from which design solutions shall be determined via synthesis (6.5). Prior to synthesis, the functional architecture should be verified to assure that it meets the requirements of the validated requirements baseline.

6.4 Functional verification

The enterprise shall conduct the functional-verification process to assess the completeness of the functional architecture in satisfying the validated requirements baseline and to produce a verified functional architecture for input to synthesis. The tasks that define functional verification are identified in Figure 13.

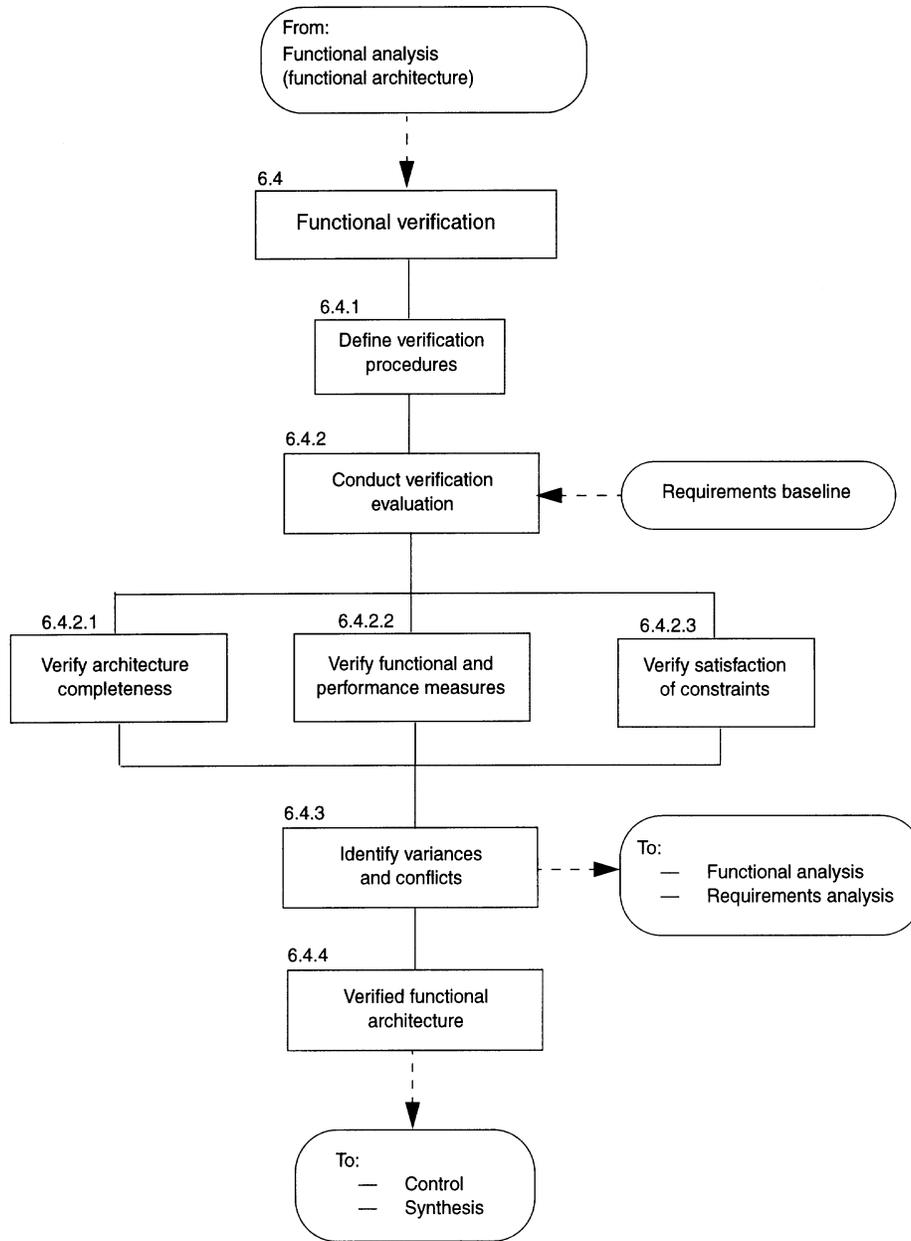


Figure 13—Functional verification

6.4.1 Define verification procedures

The enterprise defines the procedures for verifying the established functional architecture.

6.4.2 Conduct verification evaluation

The enterprise conducts defined procedures to verify that each requirement and constraint described by the established functional architecture is upward traceable to the validated requirements baseline, and that all top-level system requirements and constraints recorded in the requirements baseline are downward traceable to the functional architecture.

6.4.2.1 Verify architecture completeness

The enterprise verifies that system functional and operational requirements included in the requirements baseline are traceable to the functional architecture.

6.4.2.2 Verify functional and performance measures

The enterprise verifies that all system-level functional and performance requirements of the requirements baseline are traceable to the established functional architecture.

6.4.2.3 Verify satisfaction of constraints

The enterprise verifies that all system-level policy, procedural, standardization, functional, and design constraints of the requirements baseline are traceable to the established functional architecture.

6.4.3 Identify variances and conflicts

The enterprise identifies variances and conflicts resulting from verification evaluation activities of task 6.4.2. When incompleteness is shown, functional analysis tasks (6.3) are repeated to correct voids. When functional architecture requirements are not upward traceable to the validated requirements baseline, it should be determined if nonrequired functions and/or performance requirements were introduced during functional analysis, or whether valid functional and/or performance requirements were derived and need to be reflected in the requirements baseline. The former requires that functional analysis (6.3) be repeated to eliminate nonrequired functions and/or performance requirements. For the latter, requirements analysis (6.1) and validation (6.2) should be repeated to produce a revised validated requirements baseline.

6.4.4 Establish verified functional architecture

The functional architecture is verified upon satisfactorily resolving the variances and conflicts identified in 6.4.3. The verified functional architecture, with rationale justifying the structure, trade-off analyses performed, and key decisions, is documented in the integrated database. This verified functional architecture is used in synthesis to generate design solutions to satisfy customer expectations and meet public acceptance as defined by the validated requirements baseline.

6.5 Synthesis

The enterprise shall perform synthesis for the purpose of defining design solutions and identifying subsystems to satisfy the requirements of the verified functional architecture. Synthesis translates the functional architecture into a design architecture that provides an arrangement of system elements, their decomposition, interfaces (internal and external), and design constraints. The activities of synthesis involve selecting a preferred solution or arrangement from a set of alternatives and understanding associated cost, schedule, performance, and risk implications. Systems analysis (6.7) is used, as necessary, to evaluate alternatives; to identify, assess, and quantify risks, and select proper risk mitigation approaches; and to understand cost, schedule, and performance impacts. As subsystem requirements are defined, the identification of the needs, requirements, and constraints for life cycle processes is completed. The tasks that define synthesis are identified in Figure 14.

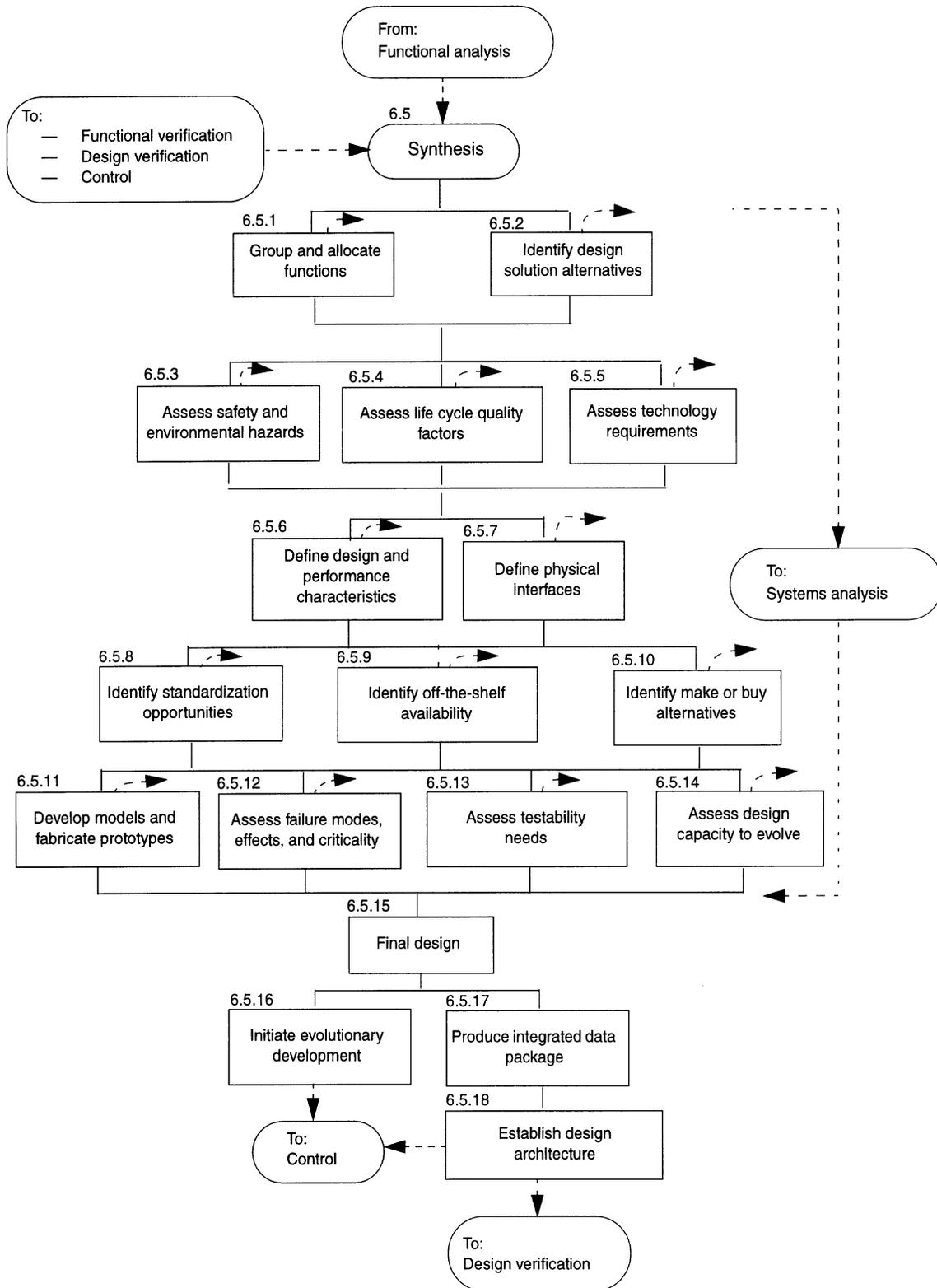


Figure 14—Synthesis process

6.5.1 Group and allocate functions

The enterprise groups common functions and subfunctions of the verified functional architecture into logical functional elements in a manner that permits their allocation to design elements. Allocation to design elements occurs when it is determined that the functional element can be accomplished by existing or newly developed items. If the functional element requires decomposition to permit its allocation, the functional decomposition is performed (6.3.2) to partition the functional element sufficiently to permit its allocation among hardware, software, and humans. Requirements traceability is established and recorded to ensure that all functions are allocated to elements of the system; each system element performs at least one function.

6.5.2 Identify design solution alternatives

The enterprise generates alternative design solutions for the functional elements identified in 6.5.1. These solutions should be composed of one or more of the following: hardware, software, material, data, facility, people, and techniques. As tasks 6.5.3 through 6.5.14 are completed, alternatives and aggregates of alternatives are analyzed to determine which design solution best satisfies allocated functional and performance requirements, interface requirements, and design constraints and adds to the overall effectiveness of the system or higher-level system. As these tasks are completed, specialty engineers work with design engineers to ensure requirements such as reliability, availability, maintainability, supportability, usability, safety, health factors, security, survivability, electromagnetic compatibility, and radio frequency management are designed as appropriate. Additionally, life cycle process requirements are identified and defined for each alternative system product solution and aggregate of solutions.

6.5.3 Assess safety and environmental hazards

The enterprise analyzes alternatives of 6.5.2, and aggregates of alternatives, to identify potential hazards to the system, humans involved in the system and supporting the system life cycle processes, or the environment. Special attention is placed on assessing safe operations of the system and assessing pollutants, hazardous wastes, or by-products associated with manufacturing, test, distribution, operation, support, training, or disposal of the system as developed to date.

6.5.4 Assess life cycle quality factors

The enterprise assesses alternatives of 6.5.2 to determine the degree to which quality factors (producibility, testability, ease of distribution, usability, supportability, trainability, and disposability) have been included in the solutions. Additionally, the enterprise assesses whether associated life cycle process needs, requirements, and constraints of the processes are identified and defined.

6.5.5 Assess technology requirements

The enterprise assesses alternatives of 6.5.2 to determine the technological needs necessary to make the design solution effective. The risks associated with the introduction of any new or advanced technologies to meet requirements should be identified and assessed. The enterprise assesses alternatives to determine the human requirements necessary to make the system effective. The tasks, roles, and jobs assigned to humans should be analyzed and assessed to determine whether or not the humans who are a part of the system have the required knowledge, skills, and abilities. Training and personnel pipelines should also be evaluated to ensure that they meet the specified requirements.

6.5.6 Define design and performance characteristics

The enterprise identifies and documents the design and performance characteristics of design alternatives. This includes the estimation or measurement of human physical and cognitive workload levels. The design characteristics and human-engineering elements associated with life cycle quality factors should be identified and assessed.

6.5.7 Define physical interfaces

The enterprise identifies and defines the physical interfaces among products, subsystems, humans, life cycle processes, and external interfaces to higher-level systems or interacting systems. Physical interfaces that impact design include communication, data, support, test, control, display, connectivity, or resource replenishment characteristics of the interaction among subsystems, the products, humans, or other interfacing systems or a higher-level system.

6.5.8 Identify standardization opportunities

The enterprise analyzes alternatives of 6.5.2 to assess whether use of standardized end items would be technologically and economically feasible.

6.5.9 Identify off-the-shelf availability

The enterprise analyzes alternatives of 6.5.2 to determine availability of an off-the-shelf item (nondevelopmental hardware or software). Each identified off-the-shelf item must be assessed to determine cost effectiveness, quantity, availability, supportability, and viability of the supplier and/or their product.

6.5.10 Identify make-or-buy alternatives

The enterprise performs economic analysis of design alternatives to support make-or-buy decisions. This analysis should address whether it is more cost effective for the enterprise to produce the design element vs. going to an established supplier.

6.5.11 Develop models and prototypes

The enterprise develops models and/or prototypes to assist in

- a) Identifying and reducing risks associated with integrating available and emerging technologies;
- b) Verifying that the design solution (made up of hardware, software, material, humans, facilities, techniques, data, and/or service) meets allocated functional and performance requirements, interface requirements, workload limitations, and constraints;
- c) Verifying that the design solution satisfies functional architecture and requirements baseline requirements.

The models, data files, and supporting documentation should be maintained, and each version of a model or data file that impacts requirements, designs, or decisions should be saved in the integrated database. Models may be digital, partial, or complete and may be hardware, software, or a combination of both, or may include human models or human-in-the-loop simulations or mock-ups for usability testing and workload measurement.

6.5.12 Assess failure modes, effects, and criticality

The enterprise assesses failure modes, the effects, and the criticality of failure for design alternatives. The hardware, software, and human elements of the design alternatives should be analyzed, and historical or test data should be applied, to refine an estimate of the probability of successful performance of each alternative. A failure modes and effects analysis (FMEA) should be used to identify the strengths and weaknesses of the design solution. For critical failures, the enterprise conducts a criticality analysis to prioritize each alternative by its criticality rating. The results of this analysis are used to direct further design efforts to accommodate redundancy and to support graceful system degradation.

6.5.13 Assess testability needs

The enterprise assesses the testability of design alternatives to determine built-in test (BIT) and/or fault-isolation test (FIT) requirements to support operational or maintenance considerations. BIT-FIT mechanisms should be provided for those elements that are normally maintained by the operators, customer, or field support engineers. BIT-FIT can be used for diagnostic operations to support lower-level maintenance actions.

6.5.14 Assess design capacity to evolve

The enterprise assesses design alternatives to determine the capacity of the design solution to evolve or be re-engineered, accommodate new technologies, enhance performance, increase functionality, or incorporate other cost-effective or competitive improvements once the system is in production or in the marketplace. Limitations that may preclude the ability of a system to evolve should be identified, and approaches analyzed and defined for resolving limitations. The enterprise should perform configuration management on products to ensure that products that have the capacity to evolve can be re-engineered cost effectively. The supportability of an evolving system may require the support process to evolve along with the product. This consideration may weigh significantly upon support funding and training requirements.

6.5.15 Finalize design

The enterprise finalizes the design for the selected alternative. The designation and description of interfaces (internal and external) among design elements are finalized.

6.5.16 Initiate evolutionary development

The enterprise initiates an evolutionary development, if necessary, for any design element for which a lesser technology solution was selected over a higher-risk technology, and for which the capacity to evolve was designed into the element and interfacing elements.

6.5.17 Produce integrated data package

The enterprise completes the drawing, schematics, software documentation, manual procedures, etc., as necessary, to document the selected design elements in an integrated data package.

6.5.18 Establish design architecture

The enterprise establishes the design architecture, appropriate to the level of development, to document the design solution and interfaces. The design architecture includes the requirements traceability and allocation matrices, which capture the allocation of functional and performance requirements among the system elements. Design architecture definitions should be stored in the integrated database, along with trade-off analysis results, design rationale, and key decisions to provide traceability of requirements up and down the architecture. Verification of the design architecture (6.6) should be accomplished to demonstrate that the architecture satisfies both the validated requirements baseline and the verified functional architecture.

6.6 Design verification

The enterprise shall perform design verification for the purpose of assuring that

- a) The requirements of the lowest level of the design architecture, including derived requirements, are traceable to the verified functional architecture; and
- b) The design architecture satisfies the validated requirements baseline.

The design verification tasks are identified in Figure 15.

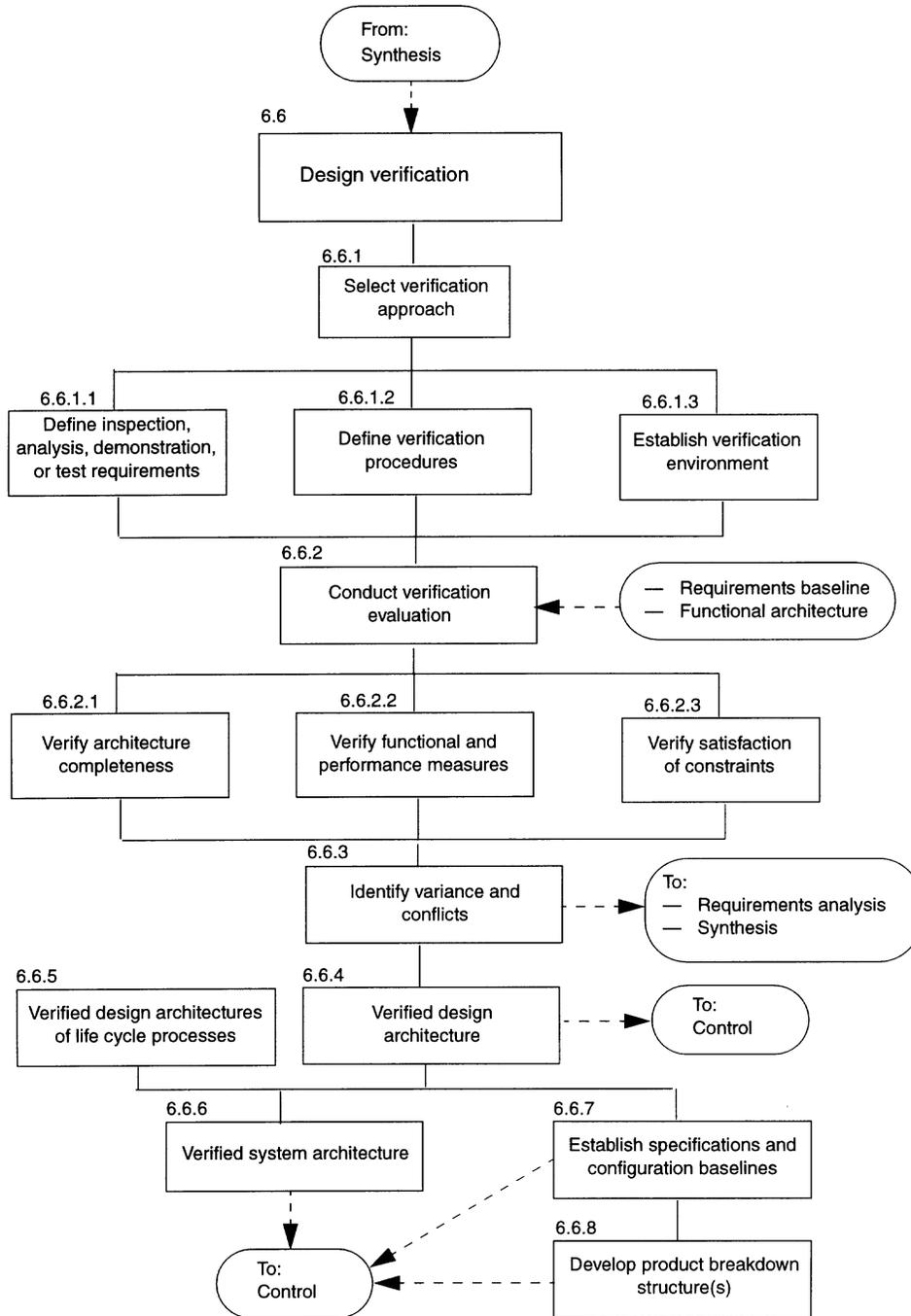


Figure 15—Design verification

6.6.1 Select verification approach

The enterprise selects the approaches for verifying the design architecture, assessing design completeness, and determining the satisfaction of MOEs and MOPs.

6.6.1.1 Define inspection, analysis, demonstration, or test requirements

The enterprise selects the appropriate verification method [inspection, analysis (including mock-ups or simulations), demonstration, or test] for evaluating whether functional and performance requirements, and design characteristic identified in the design architecture, are satisfied. A verification matrix is developed to trace the verification method(s) to requirements of the functional architecture and requirements baseline. The enterprise also selects the models or prototypes to be used, which may be partial or complete, and may or may not include humans depending on the purpose and objectives of the verification task.

6.6.1.2 Define verification procedures

The enterprise defines procedures for each verification method selected, identifies the purpose and objectives of each verification procedure, identifies the pretest and posttest actions, and defines the criteria for determining the success or failure of the procedure for planned and abnormal conditions.

6.6.1.3 Establish verification environment

The enterprise establishes the environment for methods selected and procedures defined. Environment considerations include facilities, equipment, tools, simulations, measuring devices, personnel, and climatic conditions. Environment should be checked-out prior to conducting verification.

6.6.2 Conduct verification evaluation

The enterprise conducts verification evaluation to ensure that each requirement and constraint is traceable to the verified functional architecture and that the design element solutions satisfy the validated requirements baseline. The verification results are evaluated to ensure that the behavior exhibited by the design element solutions was anticipated and satisfies requirements.

6.6.2.1 Verify architecture completeness

The enterprise verifies that

- a) Design elements descriptions are traceable to requirements of the functional architecture (upward traceability); and
- b) The requirements of the functional architecture are allocated and traceable to the design architecture.

All internal and external design interfaces should be upward and downward traceable to their source requirement.

6.6.2.2 Verify functional and performance measures

The enterprise verifies that the evaluation results from activities defined by 6.6.1 satisfy the functional and performance requirements (including human performance requirements) of the validated requirements baseline.

6.6.2.3 Verify satisfaction of constraints

The enterprise verifies that the

- a) Evaluation results from activities, as defined by 6.6.1, satisfy the constraints, including interfaces, of the functional architecture; and
- b) Constraints of the established design architecture are traceable to the validated requirements baseline.

6.6.3 Identify variances and conflicts

The enterprise identifies variances and conflicts resulting from verifying activities. When variances show incompleteness, synthesis tasks (6.5) or functional analyses tasks (6.3) are repeated to correct omissions. When evaluation results do not verify functional architecture requirements, or when design architecture requirements are not traceable to the functional architecture, it should be determined if nonrequired functions and/or performance requirements or design elements were introduced during synthesis, or whether valid functional and/or performance requirements were introduced and need to be reflected in the functional architecture. The former requires that synthesis (6.5) be repeated to eliminate nonrequired functions and/or performance requirements. For the latter variance or conflict, requirements analysis through functional verification (6.1 through 6.4) should be repeated to produce a new validated requirements baseline and verified functional architecture. When design architecture requirements are not traceable to the validated requirements baseline, it may require that synthesis be repeated to eliminate nonrequired functional and or performance requirements; or it may require that the SEP activities be repeated, as necessary, to include those missing requirements.

6.6.4 Verified design architecture

The design architecture is verified upon satisfactorily resolving the variances and conflicts identified in 6.4.3. The verified design architecture, with rationale justifying the architecture, trade-off analyses performed, and key decisions, is documented in the integrated database. This verified design architecture is used to form the specification tree for the system, and when combined with the verified life cycle process design architectures, forms the system architecture.

6.6.5 Verified design architectures of the life cycle process

The enterprise completes requirements analysis, functional analysis, and synthesis tasks to identify, define, and design architectures for life cycle processes. The enterprise performs tasks 6.1 through 6.4 to verify the design architecture for each life cycle process product. The products associated with each life cycle process are bought or made and integrated with other products related to the process or other processes, in a timely manner, to support key technical events.

6.6.6 Verified system architecture

A complete system architecture is composed of all life cycle process design architectures and product design architectures. The system architecture is verified upon satisfactorily completing the verification of the products and their life cycle process products.

6.6.7 Establish specifications and configuration baselines

After verification of the design architecture, the enterprise develops/updates product and interface specifications appropriate to the stage of development (see Clause 4) for each element of the design architecture. In addition, the enterprise develops/updates appropriate configuration baselines for each element of the design architecture. The hierarchy of specifications (product and interface) for the design architecture forms the specification tree appropriate for the stage of development (see Figure 5). The specification tree delineates specification elements for which a product must be fabricated, manufactured, bought, or coded. Specifications are stored in the integrated database and used in the next application of the SEP. The design solution for the next level of development must satisfy these specifications.

6.6.8 Develop system breakdown structure (SBS)

The enterprise develops an SBS for the system designed, including life cycle process requirements (see Figure 6). The SBS is stored in the integrated database and is used to structure and manage technical activities of the next stage of development.

6.7 Systems analysis

The enterprise shall perform systems analyses for the purpose of resolving conflicts identified during requirements analysis, decomposing functional requirements and allocating performance requirements during functional analysis, evaluating the effectiveness of alternative design solutions and selecting the best design solution during synthesis, assessing system effectiveness, and managing risk factors throughout the systems engineering effort. Systems analysis provides a rigorous quantitative basis for establishing a balanced set of requirements and for ending up with a balanced design. The tasks of systems analysis are identified in Figure 16. Even if a trade-off analysis is not done, an overall assessment of the system effectiveness shall be completed.

6.7.1 Assess requirement conflicts

The enterprise assesses conflicts among requirements and constraints identified during requirements analysis to identify alternative functional and performance requirements, where necessary. Requirements trade-off analyses and assessments are performed to identify the recommended set of requirements and constraints in terms of risk, cost, schedule, and performance impacts.

6.7.2 Assess functional alternatives

The enterprise assesses possible alternative subfunction arrangements for the decomposition of a function and for the allocation of allocable performance requirements to the subfunctions during functional analysis. Functional trade-off analyses and assessments are performed to identify the recommended set of subfunctions for each function and performance requirement allocations in terms of risk, cost, schedule, and performance impacts.

6.7.3 Assess design alternatives

The enterprise assesses potential groupings and allocations of functions from the verified functional architecture and identified design alternatives during synthesis. Design trade-off analyses and assessments are performed to identify the recommended design trade-offs in terms of risk, cost, schedule, and performance impacts.

6.7.4 Identify risk factors

The enterprise assesses requirements and constraints from requirements analysis, subfunction arrangements resulting from functional decomposition, allocation of subfunctions to functional elements, design decisions made during synthesis, and design elements of the design architecture to identify the risk factors to successful completion of the project. These evaluations should be made from an entire life cycle perspective. Identification of risk should be in a form to understand

- a) The circumstances that might lead to risk factor occurrence and the probability of occurrence;
- b) How the risk factor can be recognized if it does occur;
- c) How the risk factor affects cost, schedule, and performance.

Identified risks are prioritized based upon criticality to the successful development of the system. Acceptable levels of risk should be identified, depending on the stage of development, to provide a basis for establishing and monitoring risk reduction activities and mitigating unacceptable risks.

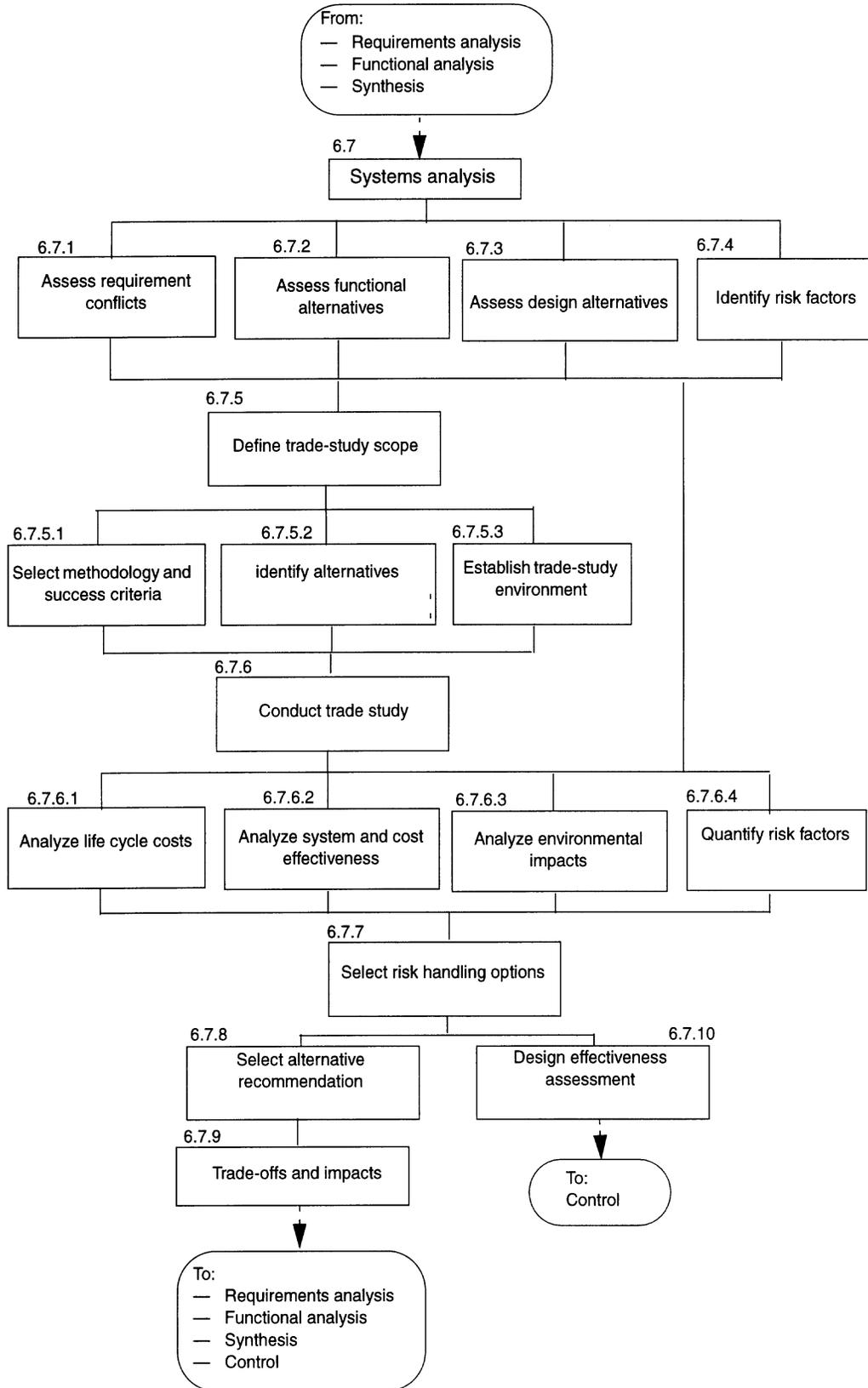


Figure 16—Systems analysis process

6.7.5 Define trade-off analysis scope

The enterprise should define the scope of the trade-off analysis to be conducted. A trade-off analysis can be

- a) Judgmental—a selection made based on the judgment of the analyst or designer, which does not require the rigor of a more formal study and for which the consequences are not too important; one alternative that is clearly superior to others; and/or time that may not be available for a more formal approach (most trade-off analyses done in accomplishing the tasks of the SEP are of the mental type);
- b) Informal—follows the same methodology of a formal trade-off analysis but is not documented as formally and is of less importance to the customer;
- c) Formal—formally conducted with results reviewed at technical reviews.

Informal and formal trade-off analysis objectives, execution, data collection requirements, schedule of activities, analysis of results, and expected outcomes need to be fully defined. Each trade-off analysis is conducted for the purpose of selecting among competing alternatives to support customer needs, system effectiveness, design-to-cost, or life cycle cost objectives within acceptable levels of risk.

6.7.5.1 Select methodology and success criteria

The enterprise selects the general approach, resources, and procedures for performing trade-studies based upon the trade-study definition, its level of importance, and availability of tools, facilities, special equipment, and related resources. The enterprise also lists the set of selection criteria, which includes factors that characterize what makes a specific alternative desirable, such as cost, schedule, performance and risk; life cycle quality factors; reuse; and size, weight, and power consumption. Adverse qualities as well as favorable qualities should be included as criteria.

6.7.5.2 Identify alternatives

The enterprise identifies and lists the viable alternative solutions to be evaluated. Each alternative should be compared with respect to completeness, and sensitivity analysis should be conducted to understand how each alternative withstands changes in the environment, technology base, or within the bounds of the evolutionary strategy.

6.7.5.3 Establish trade-study environment

The enterprise establishes metrics for each criterion that characterizes how well various alternatives satisfy the criterion. In addition, the enterprise establishes weighting factors for each criterion, which distinguish the degree of importance to the trade-off analysis definition. Models (representative or simulations) are established, when needed, to support conduct of a formal or informal trade-study. The selection of models depends on the nature of the trade-off analysis, the development stage, the type of information needed, and the characteristics of interest for an alternative. Models should be validated prior to application in a trade-off analysis.

6.7.6 Conduct trade-off analysis

The enterprise completes tasks 6.7.6.1 through 6.7.6.4, to the degree appropriate, to complete trade-off analyses for

- a) Requirements analysis to both resolve conflicts with and satisfy customer/market needs, requirements, and constraints;
- b) Functional analysis to support decomposition of functions into subfunctions and to allocate performance requirements;
- c) Synthesis to support design decisions.

Formal and informal trade-off analyses are conducted under controlled conditions to generate data pertaining to each alternative. The results of the trade-off analyses are recorded and analyzed to quantify the impact each alternative has on the system or technical effort. These results are compared against the success criteria to determine which alternative is recommended.

6.7.6.1 Analyze life cycle costs

The enterprise analyzes the costs to the enterprise and to the customer for alternative system approaches considered in a trade-off analysis or system effectiveness assessment. Life cycle cost analyses

- a) Provide requisite cost information to support trade-off analysis decisions;
- b) Provide requisite cost information for system effectiveness assessments;
- c) Include the cost of development, manufacturing, test, distribution, operations, support, training, and disposal;
- d) Include established design-to-cost goals, a current estimate of these costs, and known uncertainties in these costs;
- e) Identify the impacts on life cycle cost of proposed changes.

6.7.6.2 Analyze system and cost effectiveness

The enterprise analyzes the relationships between system effectiveness and life cycle costs to

- a) Determine performance impacts on costs;
- b) Understand value added as a function of cost;
- c) Support identification of performance objectives and requirements;
- d) Support allocation of performance to functions.

System and cost effectiveness analyses are conducted on life cycle processes of manufacturing, test, distribution, operations, support, training, and disposal to support inclusion of life cycle quality factors into system product designs, and to support the definition of functional and performance requirements for life cycle processes. The results of these analyses are used in evaluating trade-off analysis alternatives and for effectiveness assessments of the system.

6.7.6.3 Analyze safety and environmental impacts

The enterprise identifies safety and environmental impacts associated with system implementation. Applicable environmental laws and regulations should be identified, and the enterprise should ensure that these are complied with by any alternative solution. The enterprise completes an environmental impact and safety analysis to determine the impact on and by system products, and the impact of their life cycle processes on the environment, or to personnel. Use of materials, or generating by-products, which present a known hazard to the environment, are to be avoided to the extent feasible. Where not feasible, provisions must be provided for proper handling, storage, and disposal of hazardous materials or by-products. Results of these analyses influence trade-off analysis recommendations and assessments of system effectiveness.

6.7.6.4 Quantify risk factors

The enterprise quantifies the impact of identified risk factors on the system or alternative being considered based on exposure to the probability of an undesirable consequence. For system effectiveness assessments, each element of the system architecture developed to date is assessed to determine what can go wrong, and if it goes wrong, what impact it may have on the system. For trade-off analyses, risk levels assessed during life cycle cost, system and cost effectiveness, and environmental impact analyses are prioritized and reported as part of trade-off analysis recommendations.

6.7.7 Select risk-handling options

The enterprise assesses various risk-handling options to select those that may mitigate risks consistent with the current stage of development and risk-management policies set by the project. Risk, which may be reduced by lessening either the likelihood or the impact, or both, may be accepted given the cost, schedule, and performance impacts and planned mitigation approaches. An analysis of the risk-handling options should be accomplished to quantify costs and effects on the probability and impact of risk. The enterprise should select those risk-handling options that are feasible, and that reduce risks to acceptable levels with the best cost/benefit ratio. The expected remaining risks after risk-handling mitigation efforts are implemented should be identified and quantified. Throughout risk identification, quantification, and handling, integration is needed from lower levels of the system architecture up through the system level to understand cause-and-effect interactions. Risk reduction approaches and expected remaining risks are included in a risk reduction plan, which is included in trade-off analysis recommendations and effectiveness assessment reports. The complete risk reduction effort is documented in the engineering plan and integrated into the master schedule for the next stage of development, and briefed at appropriate technical reviews.

6.7.8 Select alternative recommendation

The enterprise utilizes the results of trade-off analyses and risk-reduction planning information to recommend a preferred alternative to the decision maker. The enterprise should assess the trade-off analysis to assure that the methodologies and data collection instrumentation were sufficient to support a fair and complete evaluation. Each recommendation should be presented in terms of configuration and cost, schedule, performance, and risk impact.

6.7.9 Trade-offs and impacts

The enterprise documents the recommended trade-off alternative(s) with corresponding impacts and presents the results to the appropriate decision makers within the SEP activity who are making or requesting the trade-off analysis. The final alternative selection is made based on the criteria established to judge a desirable solution. Key trade-off analysis activities, decisions, rationale, and recommendations are documented in the integrated database.

6.7.10 Design effectiveness assessment

The enterprise determines the effectiveness of the current system design based on the results of the assessments and analyses. The results of these assessments and analyses are documented in the integrated database and briefed at appropriate technical and project reviews.

6.8 Control

The enterprise shall perform the tasks of control for the purpose of managing and documenting the activities of the SEP. The tasks of control are identified in Figure 17. Outputs and test results, the planning for the conduct of the SEP activities (engineering plan, master schedule, and detail schedule), and technical plans generated by engineering specialties are controlled by the enterprise. The control tasks provide

- a) A complete and up-to-date picture of SEP activities and results, which are used in accomplishing other activities;
- b) Planning for and inputs to future applications of the SEP;
- c) Information for production, test, and customer support;
- d) Information for decision makers at technical and project reviews.

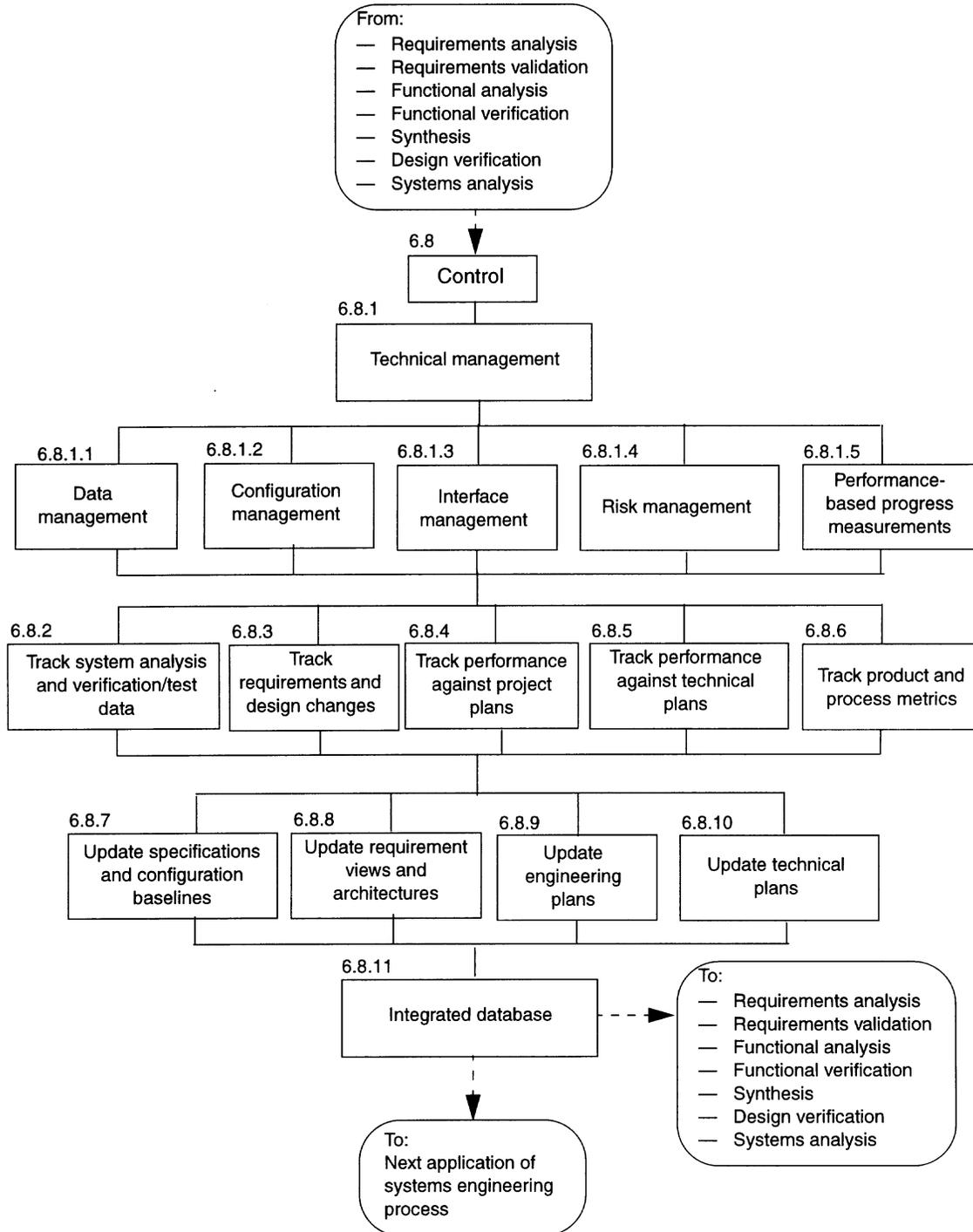


Figure 17—Control process

6.8.1 Technical management

The enterprise manages the tasks and activities of the SEP to control data generated, configuration of the design solutions, interfaces, risks, and technical progress. The enterprise needs to maintain the correct staffing, facilities, equipment, and tools; manage costs and schedules; replan development activities, as required;

coordinate technical interactions with customers; assure proper training of technical personnel and team members; measure technical progress; and coordinate the activities among the technical and business specialties needed to accomplish the systems engineering tasks of this standard.

6.8.1.1 Data management

The enterprise conducts data management to support the development, control, and delivery of required technical data throughout the technical effort. Data-management activities include setting up appropriate databases and procedures for capturing and retaining design data and schema, tools, and models. Data pertinent to the technical effort are readily accessible and should be maintained throughout the system life cycle. Safeguards are implemented to ensure data integrity and security, and to prevent inadvertent loss or modification of data. The enterprise has the responsibility to assure that the data is collected, stored, controlled, and available for proper configuration management of the evolving product designs, specifications, and baselines. Data-management and design-capture activities should be coordinated.

6.8.1.2 Configuration management

The enterprise plans and implements the functions of

- a) Identification of end items to be controlled (through specifications, interface control drawings/documents, and configuration baselines);
- b) Control of engineering changes;
- c) Status accounting;
- d) Configuration audits.

Data pertinent to configuration management are readily accessible throughout the system life cycle. The enterprise establishes and maintains configuration control boards to process, review, and approve engineering changes to end items.

6.8.1.3 Interface management

The enterprise plans and implements the functions of interface definition, interface control, interface compatibility assessments, and interface coordination. Data pertinent to interface management are readily accessible throughout the system life cycle. The enterprise establishes and maintains interface working groups to process, review, and make recommendations on approval of interface changes.

6.8.1.4 Risk management

The enterprise should conduct risk management to systematically control the uncertainty in the project's ability to meet cost, schedule, and performance objectives. The enterprise conducts that part of risk management that directly impacts the technical effort and involves risk-management preparation, risk assessment, risk-handling option assessment, and risk control. These activities are described as follows:

- a) *Risk-management preparation* includes identifying acceptable levels of risk for each stage of development; identifying potential sources of risk; identifying and evaluating risk-management tools and techniques; training team members to accomplish risk-management activities; deciding on the means by which the analysis and decisions that occur should be documented; and deciding on how risk-management information should be captured, processed, and disseminated.
- b) *Risk assessment* includes identifying and describing those circumstances which might result in adverse effects; quantifying these circumstances to determine their likelihood and potential cost, schedule, and performance effects; and ranking and integrating these risks to produce a risk assessment for each element of the SBS appropriate to the stage of development.

- c) *Risk-handling option assessment* involves evaluating risk-handling options to determine those that are feasible and that reduce risks to acceptable levels. Lower-level risk-handling options should be integrated with higher-level options that are feasible and that provide the best balance between cost, schedule, performance, and risk for the technical effort and the project.
- d) *Risk control* involves continuously assessing risk in order to provide current risk information, and to ensure that risk stays within acceptable levels.

6.8.1.5 Performance-based progress measurement

The enterprise measures, evaluates, and tracks the progress of technical efforts with the help of the master schedule, TPM, cost and schedule performance measurements, and technical reviews. The activities associated with these measurements are described as follows:

- a) The master schedule identifies tasks and activities, with associated success criteria, of an element of the SBS that should be accomplished to pass a defined technical event. A master schedule provides top-level process control and progress measurements that
 - 1) Ensure completion of required technical tasks;
 - 2) Demonstrate progressive achievements and maturity;
 - 3) Ensure that integrated, interdisciplinary information is available for decisions and events;
 - 4) Demonstrate control of cost, schedule, and performance risks in satisfying technical tasks, requirements, and objectives.
- b) TPMs, when appropriately selected, are key to progressively assessing technical progress. Each critical technical parameter should be tracked relative to time, with dates established as to when progress will be checked, and when full conformance will be met. Key technical parameters are measured relative to lower-level elements of the SBS by estimate, analysis, or test, and values are rolled up to the system level. TPM is also used to
 - 1) Assess conformance to requirements;
 - 2) Assess conformance to levels of technical risk;
 - 3) Trigger development of recovery plans for identified deficiencies;
 - 4) Examine marginal cost benefits of performance in excess of requirements.

The enterprise reports out-of-tolerance measurements to the project manager so that needed corrective actions may be taken.

- c) Cost and schedule performance measurements assess progress based on actual cost of the work performed, the planned cost of the work performed, and the planned cost of the work scheduled. Calculated cost and/or schedule variances quantify the effect of problems being experienced. Cost and schedule performance measurements are integrated with TPMs to provide current cost, schedule, and performance impacts, and to provide an integrated corrective action to variances identified.
- d) Technical reviews are conducted at the completion of an application of the SEP and/or end of a stage of development to assure that all master schedule criteria have been met; assess development maturity to date and the product's ability to satisfy requirements; assure traceability of requirements and validity of decisions; and assess risks related to investment needed for, and preparation for, the next stage of the life cycle. Clause 5 describes the reviews and audits required and their relationship to life cycle activities.

6.8.2 Track systems analysis and test data

The enterprise collects, analyzes, and tracks data from systems analyses to document activities, rationale, recommendations, and impacts, and from tests to document results, variances, and follow-up activities.

6.8.3 Track requirement and design changes

The enterprise collects and sorts data to track requirement and design changes and to maintain traceability of change source, processing, and approval.

6.8.4 Track progress against project plans

The enterprise collects and sorts data reflecting plan activities and tracks progress against the engineering plan, master schedule, and detail schedule. Deviations from plans and needed changes shall be requested in advance, and shall be undertaken only when approved.

6.8.5 Track progress against engineering plans

The enterprise collects and sorts data reflecting plan activities and tracks progress against engineering and technical plans to determine deviations from plans and needed changes, and to document changes, decisions, and accomplishments.

6.8.6 Track product and process metrics

The enterprise collects, analyzes, and tracks product and process metrics to

- Determine technical areas requiring project management attention;
- Determine the degree of customer satisfaction and public acceptance;
- Provide cost and schedule estimates for new products and provide faster response to customers.

Metrics are collected, tracked, and reported at preestablished control points during each stage of development to enable the

- Establishment of a quality system and achievement of efficient use of resources;
- Overall system quality and productivity evaluation;
- Comparison to planned goals and targets;
- Early detection of problems;
- Benchmarking of the SEP.

6.8.7 Update specifications and configuration baselines

The enterprise updates specifications and configuration baselines to reflect all changes approved by the configuration control board. The original configuration baseline, with approved changes, provides the basis for continuing technical efforts.

6.8.8 Update requirements views and architectures

The enterprise updates requirements views and the functional, design, and system architectures to reflect changes brought about by a customer, systems analysis, validation and verification deviation, or management decision. The updated requirements baseline or functional, physical, or system architecture is used for continuing SEP activities.

6.8.9 Update engineering plans

The enterprise should update the engineering plans to reflect changes brought about by a customer, systems analysis, cost or schedule deviation, or management decision. Updates should include SEP and scheduling planning activities for the next stage of the life cycle.

6.8.10 Update technical plans

The enterprise should update the technical plans to reflect changes brought about by a customer, systems analysis, plan activity deviation, or management decision. Updates should include technical planning activities for the next stage of the life cycle.

6.8.11 Integrated database

The enterprise prepares and maintains a database to store all pertinent data and information from tasks 6.8.1 through 6.8.10. This database is a repository for all information used and generated by the SEP and describes the current state of the system development and its evaluation. An electronic medium is preferred over a manual storage and retrieval database. It should be a sharable database so that all development team participants have access to the data/information needed. The integrated database needs to be accurate, unambiguous, secure, survivable, easily accessible by authorized users, and complete. The enterprise should regularly back up the database using appropriate media to enable recovery from disaster, failure of equipment or media, or accidental deletion of data.

Annex A

(informative)

The role of systems engineering within an enterprise

A.1 The systems engineering process (SEP)

The SEP provides a focused approach for product development that attempts to balance all factors associated with product life cycle viability and competitiveness in a global marketplace. This process provides a structured approach for considering alternative design and configurations. Figure A.1 provides a view of the SEP and its role within enterprise and external environments to establish a system design associated with a product offering.

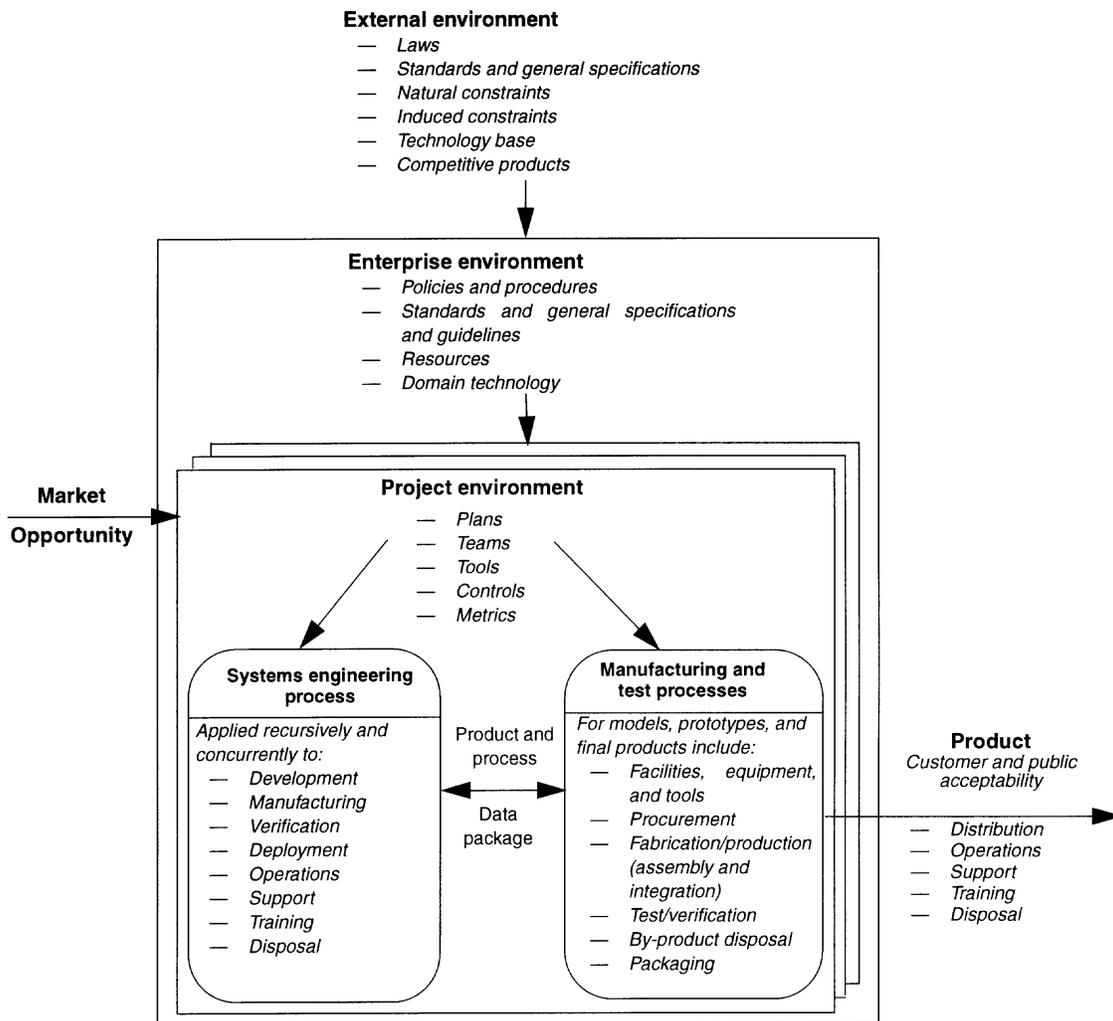


Figure A.1—Systems engineering environment within an enterprise

The SEP is applied recursively, one level of development at a time. Initially, it is applied to identify the best concept, or approach, to satisfy the market opportunity. This could be a concept for a totally new product/system or a concept for making an incremental improvement to an already established product. The second

application of the process adds value to the concept by fully describing the product/system definition and establishing a configuration baseline. This application provides the basis for accomplishing the more detailed engineering development of subsystems, components, and elements of a total system; or the appropriate parts of an established product undergoing incremental improvement during the next application.

A.2 Systems engineering internal to an enterprise

The intent of the portrayal in Figure A.1 is to represent systems engineering as the total technical effort responsible for establishing the product design, as well as the concurrent establishment of the development, test, manufacturing, support, operations, training, distribution, and disposal processes. Thus, the term *systems engineering* implies that a total system perspective should be applied to the development of a system. An enterprise should be concerned with two key processes in order to commercialize a product and achieve customer and public acceptance. The SEP establishes the design of the product and its supporting life cycle infrastructure. The manufacturing process transforms raw material, parts, etc. and assembles them into finished products in accordance with integrated data package specifications and instructions. The following subclauses discuss some of the general concepts addressed by Figure A.1.

A.2.1 Market opportunity

The enterprise may recognize an opportunity that arises from market research, research and development activities, or technology applications. In such cases, the enterprise is not responding to clearly defined customer requirements or needs, but is attempting to stimulate product acceptance through innovation. In some instances, technological advancement and the system development of this technology provide new market opportunities. Collateral material representing market research or related material pertinent to the product may be available and should establish the customer quality attributes that define a quality product offering within the marketplace. Additionally, an opportunity may arise when an enterprise is contracted to produce a product at the request of a customer. This relationship demands that the enterprise understand the customer's needs and develop a product (in a system context) to satisfy customer and public expectations.

A.2.2 New technological advances

New technologies may have a broad effect on the performance and capabilities of new products. Thus, as an input to the requirements process, new technologies should be assessed for their value to improving the product's design or capability.

A.2.3 Project environment

The project environment defines the objectives, success criteria, project milestones, and associated management priorities that will govern the integrated technical activities in support of product development. The methods by which the project is to be accomplished within the project environment should be documented in a systems integrated management plan. The efficiency and effectiveness of project integrated technical activities are enhanced by

- a) Integrated, multidisciplinary teamwork; and
- b) Appropriate computer-aided integrated tools.

A.2.4 Enterprise environment

The enterprise establishes the policies and procedures that govern project activities associated with product development. Additionally, enterprise standards and general specifications or guidelines govern development activities and product designs. These directives represent enterprise guidelines for establishing a viable product in a competitive marketplace. Enterprise management should allocate the resources available to

accomplish the project systems engineering tasks and activities in support of establishing the product design, manufacturing, test, operations, support distribution, training, and disposal processes. Enterprise activities include the training of project personnel, establishing key application technologies, and implementing the enterprise information infrastructure for control of projects. The domain technologies of the enterprise also constrain tool availability and use, design alternatives, and process solutions.

A.2.5 External environment

The external environment provides the political and social opinions or constraints that affect enterprise endeavors to commercialize new products. The enterprise should ensure that the product is designed to be compliant with applicable sociopolitical constraints. These constraints constitute the sociopolitical climate under which commercial or industrial activities are regulated and include environmental protection regulations, safety regulations, technological constraints, and other regulations established by federal and local government agencies to protect the interests of consumers. Additionally, international, government, and industry standards and general specifications constrain enterprise and project activities and design options. Competitors' products should be understood by the enterprise in order to set benchmarks for improving their product designs or make the decision not to compete in a given product area. Another constraint on product solutions is provided by the natural and induced environments in which a product will operate. The impact of these environments, as well as the impact a given product may have on these environments, establish to a large extent the public acceptability of the product in the marketplace.

A.2.6 Products

A basic interest of an enterprise is to market products that satisfy customer expectations and have general public acceptance. Acceptance includes having the applicable services of distribution, training, support, and disposal available when needed to sustain product use.

A.3 The systems engineering problem and solution space

In this context, systems engineering is responsible for the total development effort necessary to establish a product design that can be tested, manufactured, supported, operated, distributed, and disposed of. Also, the training for operation, support, distribution (installation, etc.), and disposal should be accounted for. The challenge of engineering a system to satisfy the combination of customer expectations, enterprise policies, and social, legal, and geopolitical restrictions requires a structured process for exploring options in system alternatives to ensure that a cost-effective, practical design is developed. Figure A.2 depicts the problem space that should be explored and well understood in order to begin developing a product solution.

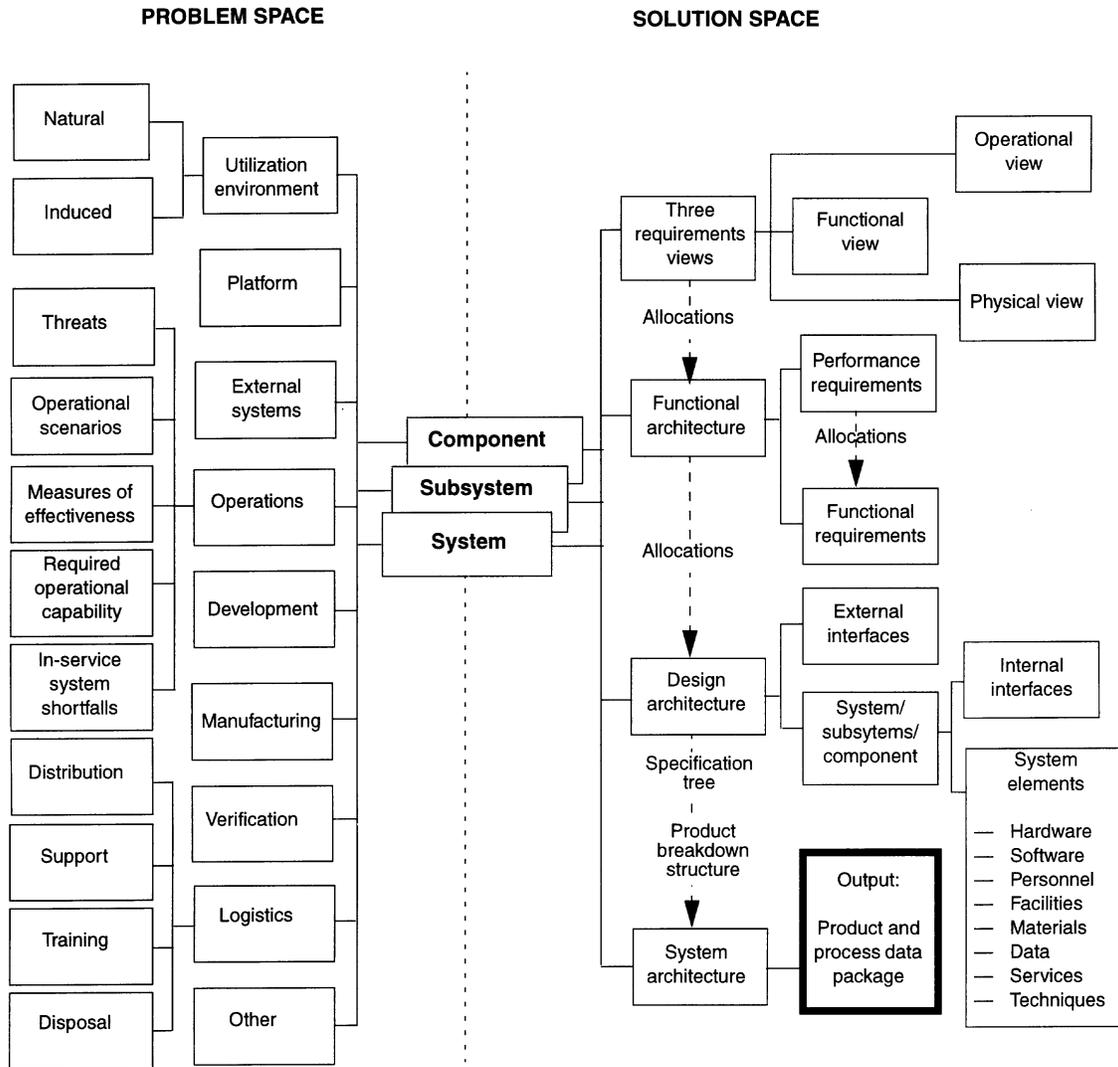


Figure A.2—Problem and solution space for systems engineering

Annex B

(informative)

The systems engineering management plan

B.1 Engineering plan template

The purpose of this engineering plan template is to provide an enterprise with a format for preparing a systems engineering management plan. Typically, a project has a project-specific engineering plan for each stage of development and uses it to guide and track systems engineering activities. The project-specific engineering plan should be compliant with project management plans; enterprise plans, capabilities, and constraints; and customer expectations.

Since each project may have unique life cycle dimensions, tolerance for risk, and need for data, the engineering plan should be tailored for each application.

B.2 Engineering plan structure

The engineering plan is a living document and needs to be structured to allow for ease of updating to reflect changes and progress throughout a stage of the life cycle. Frequently changed data may be collected in a table. Data that require high-level approval to change should be separate from that which the enterprise may change. A configuration management plan for the engineering plan should be included in the engineering plan. Information should not be duplicated in multiple sections. A simple cross reference would be helpful in appropriate sections. As a guide to a preparer, typical sections are provided in the recommended template structure. A description of what each section and subsection should contain is given below.

Title Page. Includes the words “systems engineering management plan,” the document control number for the project, organization involved, and the document title and/or applicable system. Figure B.1 provides an example of a title page with the necessary information identified.

Table of Contents. Lists the section title and page number of each titled paragraph and subparagraph. The table of contents should also list the title and page number of each figure, table, and appendix, in that order. (Page numbers in Figure B.2 represent template references only. No document length is inferred or implied.)

Section 1.0—Scope. Includes a brief description of the purpose of the system to which the engineering plan applies and a summarization of the purpose and content of the engineering plan and how its configuration will be managed.

Section 2.0—Applicable Documents. Lists all government, ISO, industry, enterprise, project, and other directive documents applicable to the conduct of the tasks within the engineering plan.

Section 3.0—Systems Engineering Process (SEP) Application. Describes the tasking/enterprise’s SEP activities as they are to be applied to the total engineering effort of the project and the organizational responsibilities and authority for systems engineering activities, including control of supplier engineering. Descriptions include the tasks needed to satisfy each accomplishment criteria identified in the master schedule and the milestones and schedules of the systems engineering detailed schedule (detail schedule) for the project. Descriptions include narratives, supplemented as necessary by graphical presentations, detailing the plans, processes, and procedures for the application of the SEP.

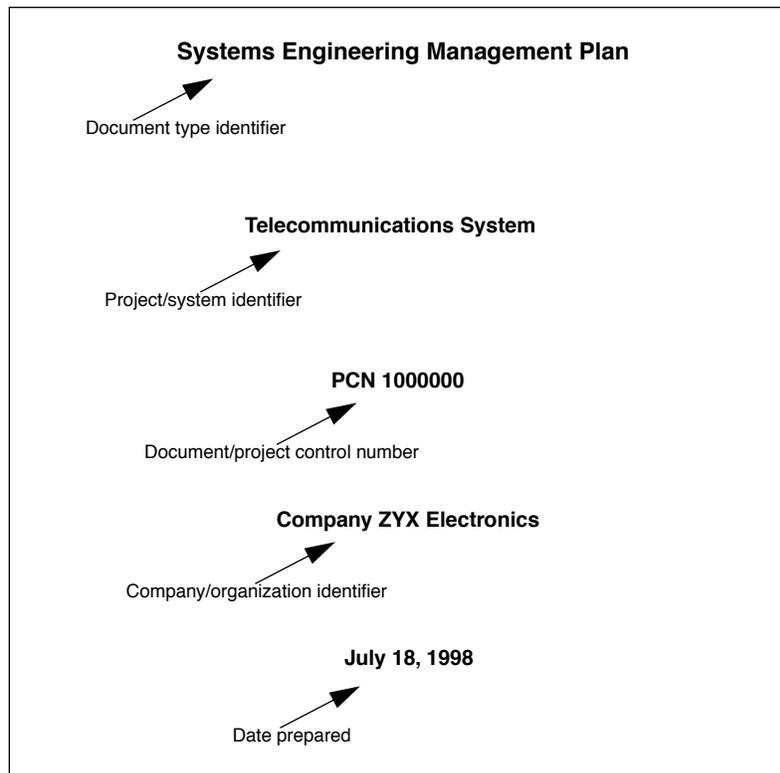


Figure B.1—Example title page

Section 3.1 Systems Engineering Process Planning. Briefly describes an overview of the key project technical objectives, deliverables and results from the process, needed process inputs, and product work breakdown structure development.

Section 3.1.1 Major Deliverables and Results. Describes in detail the major technical deliverables and results, both to the customer and internal within Company X, as a result of the SEP activities.

Section 3.1.1.1 Integrated Database. Describes the implementation of the decision database. Includes a description of how information will be captured, traced, and maintained. Provides a description of the provisioning for design-capture data/schema to include domain models (processes, technologies, etc.); product models (design prototypes—location, availability, characterization, etc.); archival data (lessons learned, past designs, empirical data); requirements, goals, and constraints; project management models (cost, schedule, and risk); integrated views, multiple views, and multidisciplinary designs and their rationale; trade-off analyses and system/cost-effectiveness analysis rationale and results; verification data; and product and process metrics.

Section 3.1.1.2 Specifications and Baselines. Describes how the generation of specifications and baselines will be documented and controlled.

Section 3.1.2 Process Inputs. Identifies the depth of detailed information needed to be able to accomplish the activities (appropriate to the level of development) of the SEP, how needed information will be acquired, and how conflicts will be resolved.

Section 3.1.3 Technical Objectives. Describes the technical objectives related to success of the project, system, and system effectiveness [e.g., customer measures of effectiveness (MOEs)]. Technical objectives may include those related to the system products and their life cycle processes.

Table of Contents		Page
1.0	Scope	5
2.0	Applicable Documents	6
3.0	Systems Engineering Process (SEP) Application	7
3.1	Systems Engineering Process Planning	7
3.1.1	Major Deliverables and Results	7
	3.1.1.1 Integrated Database	8
	3.1.1.2 Specifications and Baselines	8
3.1.2	Process Inputs	9
3.1.3	Technical Objectives	9
3.1.4	System Breakdown Structure (SBS)	10
3.1.5	Training	10
3.1.6	Standards and Procedures	11
3.1.7	Resource Allocation	11
3.1.8	Constraints	12
3.1.9	Work Authorization	12
3.2	Requirements Analysis	13
3.3	Requirements Baseline Validation	14
3.4	Functional Analysis	15
3.5	Functional Verification	15
3.6	Synthesis	16
3.7	Design Verification	17
3.8	Systems Analysis	17
3.8.1	Trade-Off Analyses	18
3.8.2	System/Cost Effectiveness Analyses	18
3.8.3	Risk Management	18
3.9	Control	19
3.9.1	Design Capture	19
3.9.2	Interface Management	19
3.9.3	Data Management	20
3.9.4	Systems Engineering Master Schedule (SEMS)	20
3.9.5	Technical Performance Measurement	20
3.9.6	Technical Reviews	20
3.9.7	Supplier Control	21
3.9.8	Requirements Traceability	21
4.0	Transitioning Critical Technologies	22
5.0	Integration of the Systems Engineering Effort	23
5.1	Organizational Structure	23
5.2	Required Systems Engineering Integration Tasks	24
6.0	Additional Systems Engineering Activities	25
6.1	Long-Lead Items	25
6.2	Engineering Tools	25
6.3	Design to Cost	25
6.4	Value Engineering	26
6.5	Systems Integration Plan	26
6.6	Interface With Other Life Cycle Support Functions	26
6.7	Safety Plan	26
6.8	Other Plans and Controls	27
7.0	Notes	27
7.1	General Background Information	27
7.2	Acronyms and Abbreviations	27
7.3	Glossary	27
FIGURES		
TABLES		
APPENDICES		
A.		
B.		

Figure B.2—Example table of contents

Section 3.1.4 System Breakdown Structure (SBS). Describes how the elements of the SBS will be developed. The relationship of the specification tree and the drawing tree with the elements of the SBS and how the system products and their life cycle processes will be related should be explained. This section should describe for each element of the SBS the methods for development and control of work packages; development of planning packages and their conversion to work packages; sizing of work packages; resource use, including integrated product teams (IPTs); traceability of changes; cost reporting and its integration to scheduling and critical path identification; and configuration management.

Section 3.1.5 Training. Identifies both internal and external (suppliers/customers) training needed. Includes analysis of performance or behavior deficiencies or shortfalls, required training to remedy, and schedules to achieve required proficiencies.

Section 3.1.6 Standards and Procedures. Describes major standards and procedures that the project will follow. Incorporates implementation of standardization tasking into pertinent sections of the SEP.

Section 3.1.7 Resource Allocation. Describes the method of resource allocation to technical tasks. Includes resource-requirements identification, procedures for resource control, and reallocation procedures.

Section 3.1.8 Constraints. Describes major constraints on the project. Includes those things the project cannot or will not do. Also includes funding, personnel, facilities, manufacturing capability/capacity, critical resources, or other constraints.

Section 3.1.9 Work Authorization. Describes the method by which work to be performed is authorized within the project. Also describes the method by which changes to work efforts will be authorized.

Section 3.2 Requirements Analysis. Documents the approach and methods for analysis of system product uses; utilization environments; inherent human limitations and capabilities; performance expectations; and design constraints and identification of needs, requirements, and constraints related to life cycle processes. Documents the approach and methods for analysis of hardware, software, and human systems engineering (manpower, personnel, training, human engineering, and system safety). Also documents the approach and methods to be used to define the functional and performance requirements for the following quality factors: producibility, testability, and integrated diagnostics, distributability (including packaging and handling, transportability, and installability), usability, supportability, trainability, and disposability; and for the following engineering specialty areas: reliability, maintainability, electromagnetic compatibility and electrostatic discharge, health hazards and environmental impact, system security, infrastructure support, and any other engineering specialty bearing on the determination of functional and performance requirements for the system for the appropriate level of development. Additionally, the approach and methods for evolving system products are described.

NOTE—Some areas may impact requirements analysis only after synthesis efforts identify implementation solution alternatives. Some of the descriptive information may be more appropriately covered under other SEP activities.

Section 3.3 Requirements Baseline Validation. Includes the approach and methods to validate that the requirements baseline established from requirements analysis is both upward and downward traceable to customer expectations, project and enterprise constraints, and external constraints.

Section 3.4 Functional Analysis. Includes a description of the approach and methods planned to determine lower-level functions, to allocate performance and other limiting requirements to lower-level functions, to define functional interfaces, and to define the functional architecture. Approaches and methods for the quality factors and engineering specialty areas in Section 3.2 are also defined.

Section 3.5 Functional Verification. Includes a description of the approach and methods planned to verify that the functional architecture established from functional analysis is both upward and downward traceable to the validated requirements baseline.

Section 3.6 Synthesis. Includes the approach and methods to transform the functional architecture into a design architecture (hardware, software, and humans to support the system life cycle), to define alternative system concepts, to define physical interfaces, and to select preferred product and process solutions. Describes how requirements are converted into detailed design specifications for hardware, software, human engineering, manpower, personnel, safety, training, and interfaces. Approaches and methods for the engineering areas, quality factors, and engineering specialty areas in Section 3.2 are also defined. In addition, nondevelopmental items and parts control are included.

Section 3.7 Design Verification. Includes a description of the approach and methods planned to verify that the design architecture, established from synthesis, is both upward and downward traceable to the functional architecture and satisfies the requirements of the validated requirements baseline; and supports baseline of configurations and specifications (including the human engineering, manpower, personnel, safety, and training specification).

Section 3.8 Systems Analysis. Includes an overview of the approach and methods planned to be utilized to arrive at a balanced set of requirements, and a balanced functional and design architecture to satisfy those requirements and control the level of development dependent outputs of the SEP. Provides an overview of the specific systems analysis efforts needed (including hardware, software, and human allocation analysis). Includes methods and tools for trade-off analyses, systems and cost effectiveness analyses, and risk management.

Section 3.8.1 Trade-off Analyses. Describes the studies planned to make trade-offs among stated requirements; design; project schedule; functional and performance requirements; function; task; and decision allocation between human, software, and hardware and life cycle/design to cost. Describes the use of criteria for decision-making and trade-off of alternative design solutions. Includes a description of technical objectives, criteria and weighting factors, and utility curves as applicable. Also describes the methods and tools planned to be used and interfaces with the integrated database.

Section 3.8.2 System/Cost Effectiveness Analyses. Describes the implementation of system and cost effectiveness analyses to support the development of life cycle balanced products and processes and to support risk management. Describes the MOEs, how they interrelate, and criteria for the selection of measures of performance (MOPs) to support the evolving definition and verification of the system. Includes description of the overall approach for system/cost-effectiveness analysis as well as manufacturing analysis; verification analysis; distribution analysis; operational analysis; human engineering, manpower, personnel, and training analysis; usability analysis; supportability analysis; safety, health hazards, and environmental analysis; and life cycle cost analysis. Describes how analytical results will be integrated.

Section 3.8.3 Risk Management. Describes the technical risk program, including the approach, methods, procedures, and criteria for risk assessment (identification and quantification), selection of the risk-handling options, and integration into the decision process. Also describes the risks associated with the development and verification requirements. Identifies critical risk areas. Describes plans to minimize technical risks (additional prototyping, technology and integration verification, and evolutionary system development). Identifies risk control and monitoring measures including special verifications, technical performance measure parameters, and critical milestones/events. Describes the method of relating technical performance measurement (TPM), the master schedule, and the detail schedule to cost and schedule performance measurement, and the relationship to the SBS.

Section 3.9 Control. Provides an overview of plans for design capture, interface management, data management, event-based scheduling, calendar-based scheduling, TPM, technical reviews, supplier control, and requirements traceability.

Section 3.9.1 Design Capture. Describes the approach and methods planned to manage the system definition (configuration) of identified system products and the related life cycle processes for manufacturing, verification, distribution, support, training, and disposal. Includes a description of change management, configuration

control procedures, and baseline management. Describes the design record for alternatives, trade-off analyses, decisions/conclusions, and lessons learned.

Section 3.9.2 Interface Management. Describes the approach and methods planned to manage the internal interfaces appropriate to the level of development to ensure that external interfaces (external to the project or at a higher level of the functional or design architecture) are managed and controlled. Includes description of change management and the interrelationship with configuration control procedures.

Section 3.9.3 Data Management. Describes the approach and methods planned to establish and maintain a data management system and the interrelationship with the design-capture system and decision database. Includes descriptions of how and which technical documentation will be controlled and the method of documentation of project engineering and technical information. Plans for security and preparation of deliverable data will also be described.

Section 3.9.4 Systems Engineering Master Schedule (SEMS). Describes the critical path methodology and criteria for event transition used to derive the master schedule and supporting systems engineering detailed schedule (detail schedule) and their structure. Includes a description of the approach and methods planned to update and maintain both the master schedule and the detail schedule.

Section 3.9.5 Technical Performance Measurement. Describes the approach and methods to identify, establish, and control key technical parameters (limited to those that are critical and/or identified by the customer). Descriptions include the thresholds, methods of measuring and tracking, update frequencies, level of tracking depth, and response time to generate recovery plans and planned profile revisions. Described parameters include identification of related risks. Describes the relationship between the selected parameter and lower-level parameters that must be measured to determine the critical parameter achievement value, which is depicted in the form of tiered dependency trees and reflects the tie in to the related system performance requirement (critical parameter). Includes definition of the correlation of each parameter in the dependency tree to a specific SBS element.

Section 3.9.6 Technical Reviews. Describes the technical reviews and/or audits (system, subsystem, component, and life cycle process) applicable to the level(s) of development covered by the engineering plan. Describes the approach and procedures planned to complete identified reviews and/or audits. The tasks associated with the conduct of each review, including responsibilities of personnel involved and necessary procedures (e.g., action item closeout procedures) are described. Includes a description of how conformance with the tasking activity engineering plan/master schedule and/or this engineering plan and enterprise master schedule will be determined, how the discrepancies identified as not meeting engineering plan/master schedule requirements will be handled, and how system products and related life cycle processes assessed to have a moderate to high risk of conformance will be addressed prior to conducting the review.

Section 3.9.7 Supplier Control. Describes the technical control of suppliers and vendors. Includes the approach and methods to flow-down requirements, manage interfaces, control quality, build long-term relationships, and assure participation on IPTs.

Section 3.9.8 Requirements Traceability. Describes how requirements traceability will be implemented. Includes the traceability between SEP activities, SBSs, and correlation, as pertinent, with the master schedule and the detail schedule. Describes the interrelationship of requirements traceability with data management and the integrated database.

Section 4.0—Transitioning Critical Technologies. Describes the approach and methods for identifying key technologies and their associated risks, and the activities and criteria for assessing and transitioning critical technologies from technology development and demonstration projects internal to the enterprise or from suppliers or other sources. Describes how alternatives will be identified and selection criteria established to determine when and which alternative technology will be incorporated into the product when moderate- to high-risk technologies are assessed, as required, to meet functional and performance requirements. Describes the planned method for engineering and technical process improvement, including procedures for establish-

ing an evolutionary system development to enable an incremental improvement approach for system products as technologies mature, or for evolution of the system.

Section 5.0—Integration of the Systems Engineering Effort. Describes how the various inputs into the systems engineering effort will be integrated and how integrated product teaming will be implemented to integrate appropriate disciplines into a coordinated systems engineering effort that meets cost, schedule, and performance objectives. Provides a brief description of the approach and methods planned to assure integration of the engineering specialties to meet project objectives.

Section 5.1 Organizational Structure. Describes how the organizational structure will support teaming. Describes the composition of teams organized to support a specific element of the SBS. Also describes major responsibilities and authority of team members by name, and includes present and planned project technical staffing. Includes planned personnel needs by discipline and performance level, human resource loading, and identification of key personnel.

Section 5.2 Required Systems Engineering Integration Tasks. Describes the approach and methods for systems engineering integration tasks such as technology verification, process proofing, fabrication of engineering test articles, development test and evaluation, implementation of software designs for system products, and customer and supplier engineering and problem solving support. This description includes an articulation of the required support team.

Section 6.0—Additional Systems Engineering Activities. Contains a brief description of other areas not specifically covered in Sections 1.0 through 5.0, but essential for planning a total systems engineering effort. Includes a brief description of additional systems engineering activities essential to successfully engineering a total system solution.

Section 6.1 Long-lead Items. Describes the long-lead items that affect the critical path of the project.

Section 6.2 Engineering Tools. Describes the systems engineering methods and tools that are planned to be implemented on the program to support systems engineering. Identifies those tools to be acquired and training requirements.

Section 6.3 Design to Cost. Describes the design-to-cost planning and how cost will be implemented and controlled as a design parameter.

Section 6.4 Value Engineering. Describes the approach and methods planned to address value engineering throughout the development cycle.

Section 6.5 Systems Integration Plan. Describes the approach and methods by which the system is assembled and integrated.

Section 6.6 Interface with other life cycle support functions. Describes the approach and methods to assure compatibility with other life cycle support functions consistent with project and enterprise plans.

Section 6.7 Safety Plan. Describes the approach and methods for conducting safety analysis and assessing the risk to operators, the system, the environment, or the public.

Section 6.8 Other Plans and Controls. Describes the approach and methods for any other plans and controls designated the tasking activity or which the enterprise system architect, systems engineer, or system integrator will use.

Section 7.0—Notes. Contains any general information that aids in understanding the engineering plan (e.g., background information; alphabetical listing of all acronyms, abbreviations, and their meanings, as used in the engineering plan; and glossary of terms used). Explains which of the items in this section are mandatory or are provided for general information.

Section 7.1 General Background Information. Provides background information that will help the implementers and managers of the activities and tasks of this engineering plan better understand and accomplish their responsibilities.

Section 7.2 Acronyms and Abbreviations. Provides an alphabetical list of acronyms and abbreviations, and their meanings.

Section 7.3 Glossary. Provides an alphabetical listing of key terms and their applied meaning within the context of this engineering plan.

Appendices. Appendices are included, as necessary, to provide information published separately for convenience in document maintenance. Included would be charts and proprietary data applicable to the systems engineering efforts required in the engineering plan. Also included as an appendix would be a summary of technical plans associated with the project. Each appendix should be referenced in one of the sections of the engineering plan where data would normally have been provided.

Annex C

(informative)

Bibliography

[B1] IEEE/EIA 12207.0–1996, IEEE/EIA Standard—Industry Implementation of ISO/IEC 12207: 1995, Standard for Information Technology—Software life cycle processes.⁴

[B2] ISO 9001: 1994, Quality systems – Model for quality assurance in design, development, production, installation, and servicing.⁵

⁴This publication is 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/>).

⁵ISO publications are available from the ISO Central Secretariat, Case Postale 56, 1 rue de Varembe, CH-1211, Genève 20, Switzerland/Suisse (<http://www.iso.ch/>). ISO publications are also available in the United States from the Sales Department, American National Standards Institute, 11 West 42nd Street, 13th Floor, New York, NY 10036, USA (<http://www.ansi.org/>).