INTERNATIONAL STANDARD

ISO/IEC 26702

IEEE Std 1220-2005

First edition
2007-07-15

Systems engineering — Application and management of the systems engineering process

Ingénierie des systèmes —- Application et management du processus d'ingénierie des systèmes

Reference number
ISO/IEC 26702:2007(E)
IEEE
Std 1220-2005

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

ISO
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

IEEE Standards

IEEE Std 1220™-2005
(Revision of
IEEE Std 1220-1998)

1220™

IEEE Standard for Application and Management of the Systems Engineering Process

IEEE Computer Society

Sponsored by the
Software and Systems Engineering Standards Committee

IEEE
3 Park Avenue, New York, NY10016-5997, USA

9 September 2005

Print: SH95334
PDF: SS95334

IEEE Std 1220™-2005
(Revision of
IEEE Std 1220-1998)

IEEE Standard for Application and Management of the Systems Engineering Process

Sponsor

Software and Systems Engineering Standards Committee
of the
IEEE Computer Society

Approved 20 March 2005

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: acquire, analysis, architecture, building block, design, development, component, hardware, interface, life cycle processes, software, supplier, synthesis, system, system life cycle, systems engineering, technical management, validation, verification

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

Print: ISBN 0-7381-4691-9 SH95334
PDF: ISBN 0-7381-4692-7 SS95334

International Standard ISO/IEC 26702:2007(E)

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 26702 was prepared by IEEE (as IEEE Std 1220-2005) and was adopted, under a special “fast-track procedure”, by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software and systems engineering*, in parallel with its approval by national bodies of ISO and IEC. The IEEE Computer Society will cooperate in the maintenance of this International Standard as a Category A liaison to SC 7.

International Organization for Standardization/International Electrotechnical Commission
Case postale 56 • CH-1211 Genève 20 • Switzerland

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. The IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and serve without compensation. While the IEEE administers the process and establishes rules to promote fairness in the consensus development process, the IEEE does not independently evaluate, test, or verify the accuracy of any of the information contained in its standards.

Use of an IEEE Standard is wholly voluntary. The IEEE disclaims liability for any personal injury, property or other damage, of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, or reliance upon this, or any other IEEE Standard document.

The IEEE does not warrant or represent the accuracy or content of the material contained herein, and expressly disclaims any express or implied warranty, including any implied warranty of merchantability or fitness for a specific purpose, or that the use of the material contained herein is free from patent infringement. IEEE Standards documents are supplied "**AS IS**."

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.

In publishing and making this document available, the IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity. Nor is the IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing this, and any other IEEE Standards document, should rely upon the advice of a competent professional in determining the exercise of reasonable care in any given circumstances.

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 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. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position, explanation, or interpretation of the IEEE.

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

Secretary, IEEE-SA Standards Board
445 Hoes Lane
Piscataway, NJ 08854
USA

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

IEEE Introduction

This introduction is not part of IEEE Std 1220-2005, IEEE Standard for Application and Management of the Systems Engineering Process.

History

IEEE Std 1220 was initially published in January 1995 as a trial-use standard. After the two-year trial-use period, the document was revised and balloted in 1998 for full publication in 1999.

The International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) formed a joint body to collaborate in fields of mutual interest for worldwide standardization efforts. ISO/IEC established a joint technical committee for international standards development, ISO/IEC JTC 1, Information Technology. In parallel with IEEE Std 1220 developments, ISO/IEC JTC 1, Subcommittee 7 (SC7), Software and Systems Engineering, began generation of ISO/IEC 15288:2002 [B3].[a] ISO/IEC 15288:2002 provides a framework based on a broad set of processes that an organization or project may employ to perform or manage the stages of a system's life cycle. ISO/IEC 15288:2002 supports the full life cycle of systems—from conception through retirement—as well as the acquisition and supply of systems.

It is the intent of ISO/IEC 15288:2002 to establish standard life cycle process descriptions suitable for most man-made systems. As such, the processes and terminology of the standard are defined at an appropriately high level of abstraction. ISO/IEC 15288:2002 does not prescribe, nor provide, detailed system engineering process definitions or methods and procedures to address detail process requirements derived from the application of this standard. ISO/IEC TR 19760:2003 [B4], the companion guide for ISO/IEC 15288:2002, lists several standards, including IEEE Std 1220-1998, that cover engineering disciplines at a lower-tier level and are suitable for implementation with ISO/IEC 15288:2002.

This revision of IEEE Std 1220-1998 is a result of an ongoing harmonization of the standards of the IEEE Computer Society's Software and Systems Engineering Standards Committee (S2ESC) and the corresponding international standards committee, ISO/IEC JTC1/SC7. This initial alignment of IEEE Std 1220-1998 with ISO/IEC 15288:2002 was developed in cooperation with ISO/IEC JTC1/SC7 and included participation of the International Council on Systems Engineering (INCOSE). The next step towards harmonization of these two standards would be the submission of IEEE Std 1220-2005 for a "fast-track" ballot with ISO/IEC JTC1/SC7 followed by a coordinated revision.

The IEEE Computer Society has embraced the top-level framework provided by ISO/IEC 15288:2002 and has adopted ISO/IEC 15288 as IEEE Std 15288™-2004. The IEEE Computer Society offered to align IEEE Std 1220-1998 with ISO/IEC 15288:2002 to facilitate the joint use of the two standards to manage system engineering efforts. The purpose of this revision of IEEE Std 1220-1998 is to identify key similarities and differences in the two standards and demonstrate how they can be used together while minimizing the impact of ISO/IEC 15288:2002 on current IEEE Std 1220-1998 users who may not employ ISO/IEC 15288:2002.

The key differences between this version of the standard, IEEE Std 1220-2005, and the 1998 version are as follows:

a) Inclusion of explanations regarding key differences between IEEE Std 1220-1998 and ISO/IEC 15288:2002 in areas such as terminology and structure

b) Minimal adjustments to some IEEE Std 1220-1998 terms and definitions for alignment with ISO/IEC publication requirements

c) Clarification of the distinction between requirements and recommendations of the standard

[a] The numbers in brackets correspond to those of the bibliography in Annex D.

d) Update of the conformance clause for alignment with IEEE standards style and rules

Most of the IEEE Std 1220-1998 content remains the same in this version. Explanations to facilitate use of IEEE Std 1220 with ISO/IEC 15288:2002 are contained in a new Annex C.

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 standard 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 and project 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 stakeholder 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 stakeholders, 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.

Notice to users

Errata

Errata, if any, for this and all other standards can be accessed at the following URL: http://standards.ieee.org/reading/ieee/updates/errata/index.html. Users are encouraged to check this URL for errata periodically.

Interpretations

Current interpretations can be accessed at the following URL: http://standards.ieee.org/reading/ieee/interp/index.html.

Patents

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 or patent applications for which a license may be required to implement an IEEE standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

Participants

At the time this standard was completed, the IEEE 1220 Revision Working Group had the following membership:

Teresa (Terry) Doran, *Chair and Editor*
Ken Crowder, *ISO/IEC JTC1/SC7 Liaison*
James Moore, *IEEE CS Liaison to ISO/IEC JTC1/SC7*
Lorraine Pajerek, *INCOSE Organizational Representative*

Werner Altmann
Jim Armstrong
Stuart Arnold
Randy Case
John Clark
Paul Croll
Alain Faisandier
Kevin Forsburg
Rich Harwell
Mark Henley
Ron Kohl
Jerome (Jerry) Lake
Jean-Philippe Lerat
Elizabeth Lotsu
John H. Mee
Frederick I. Moxley
John Napier
Steve Olson
Robert Peterson
Ken Ptack
Karen Richter
Garry Roedler
Robert J. Schaaf
Richard Schmidt
Robert Skalamera
Thomas Starai
Mark Wilson
Matthew Young

The following members of the individual balloting committee voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

Edward Bartlett
Richard Biehl
Stephen Blanchette
Juris Borzovs
Bruce Bullock
Joseph Butchko
Dino Butorac
Keith Chow
Antonio M. Cicu
Paul Croll
Gregory Daich
Geoffrey Darnton
Taz Daughtrey
Maulik Dave
Perry DeWeese
Dr. Guru Dutt Dhingra
Teresa (Terry) Doran
Einar Dragstedt
Scott Duncan
Clint Early, Jr.
Christof Ebert
William Eventoff
John Fendrich
Yaacov Fenster
Roger Fujii
Jean-Denis Gorin
Lewis Gray
Michael Grimley
Mark Heinrich
John Horch
Peeya Iwagoshi
Thomas M. Kurihara
Joerg Kampmann
Piotr Karocki
Ron Kenett
Carol Long
Yuhai Ma
Joseph Marshall
Jacques Mathot
James Moore
Dennis Nickle
Lou Pinto
Gerald Radack
Annette Reilly
David Rockwell
Garry Roedler
Helmut Sandmayr
James Sanders
Robert J. Schaaf
Carl Singer
Mitchell Smith
Luca Spotorno
Thomas Starai
David Walden
John Walz
Oren Yuen
Li Zhang

The following organizational representative voted on this standard:

Lorraine Pajerek, INCOSE

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

GEIA
NDIA

When the IEEE-SA Standards Board approved this standard on 20 March 2005, it had the following membership:

Also included are the following nonvoting IEEE-SA Standards Board liaisons:

Michelle D. Turner
IEEE Standards Project Editor

Contents

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 stakeholder needs, requirements, and constraints into a system solution. This standard is intended to guide the development of systems for commercial, government, military, and space applications. The information applies to a project 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 that 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 the following:

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.

This standard, IEEE Std 1220-2005, may be used in conjunction with ISO/IEC 15288:2002 [B3].[1] This standard generally prescribes more detailed systems engineering process and management requirements that complete or complement the process activities described in ISO/IEC 15288:2002. However, ISO/IEC 15288:2002 provides additional process definition and guidance that supports life cycle model definition and application of the systems engineering process across a system's life cycle.

1.3 How to use this standard

1.3.1 Conformance

Normative provisions of this standard, which are indicated by a "shall" statement, are requirements to claim conformance to this standard. Provisions that are indicated by a "should" statement are recommendations. An enterprise that desires to claim conformance with this standard demonstrates conformance by defining and implementing procedures for accomplishing all normative provisions.

1.3.2 Recommendations and tailoring

The enterprise should also incorporate select recommended and optional provisions into their procedures and should ensure that each project within the enterprise complies with these procedures.

Clause 4 provides some normative provisions and recommendations for implementing systems engineering within an enterprise or on a project. Normative provisions of Clause 4 include the development and maintenance of enterprise policies and procedures. Enterprise policies and procedures that describe the application of the systems engineering process (SEP) throughout a project life cycle typically provide the basis for project-specific application of the enterprise's SEP. Therefore, it is expected that an enterprise would establish and maintain such policies and procedures.

Clause 6 defines the SEP, which is accomplished iteratively to define system products and life cycle processes. Thus, the initial provision that defines each subprocess (requirements analysis, functional analysis, etc.) includes a "shall" statement to ensure that an enterprise's SEP addresses each subprocess and the tasks for performance of that subprocess. The remaining subclauses, when specified within the definition of a subprocess, are recommended to provide flexibility in adapting the subprocess and the task definitions of the SEP for the purposes and typical system engineering efforts of the enterprise. Clause 6 describes the recommended approach to project tailoring of the enterprise SEP, which would be applied for any particular project iteration of the SEP as described in Clause 5.

The initial provision of Clause 5 contains a "shall" statement to ensure that the project addresses application of the SEP throughout the system life cycle. The remaining provisions under Clause 5 describe recommended and optional activities that a project commonly performs when applying the SEP during each stage of a typical system life cycle. The recommended and optional activities of Clause 5 should be considered by the project during tailoring of the enterprise SEP for any particular iteration of the SEP throughout the system life cycle. This approach provides the project with the flexibility needed to address different levels of system development and appropriate rigor of SEP application throughout various system life cycle stages.

[1] The numbers in brackets correspond to those of the bibliography in Annex D.

1.3.3 System paradigm

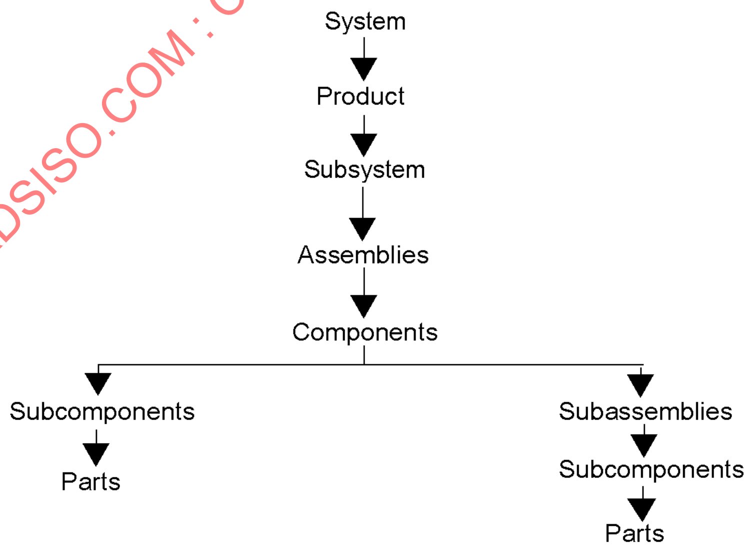
The description of the SEP, and its application throughout the life cycle, demands the use of a system paradigm to aid the presentation of this material. The terms used to support this paradigm are defined in Clause 3. As enterprises and projects gain familiarity with the paradigm, they may substitute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

1.3.3.1 Hierarchy of system elements

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 judgment call—complex components may demand the rigors of the SEP or may be well suited for component development teams.



Elements of the system may include hardware, software, and humans dependent on the system definition.

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 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.

1.3.3.2 Building block structure

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.

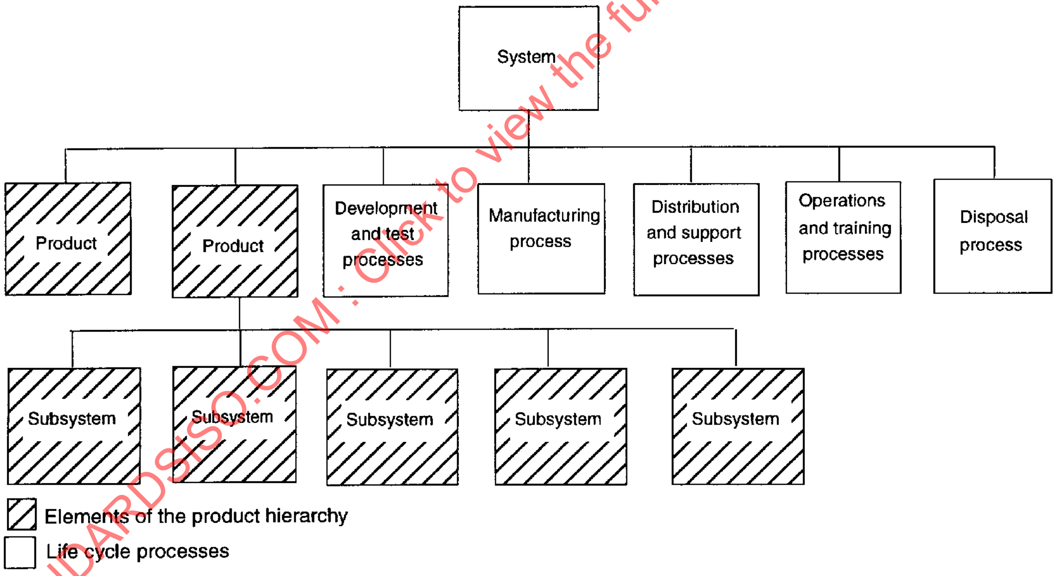

System
Product
Product
Development and test processes
Manufacturing process
Distribution and support processes
Operations and training processes
Disposal process
Subsystem
Subsystem
Subsystem
Subsystem
Subsystem
Elements of the product hierarchy
Life cycle processes

Figure 2—Basic building blocks of a system

1.3.3.3 Product and life cycle process definition

Figure 3 depicts the life cycle processes, 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 SEP 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 stakeholder 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 brass-boards, 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 stakeholder satisfaction.

d) *Distribution.* The tasks, actions, and activities to initially transport, deliver, assemble, install, test, and check out 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.

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.

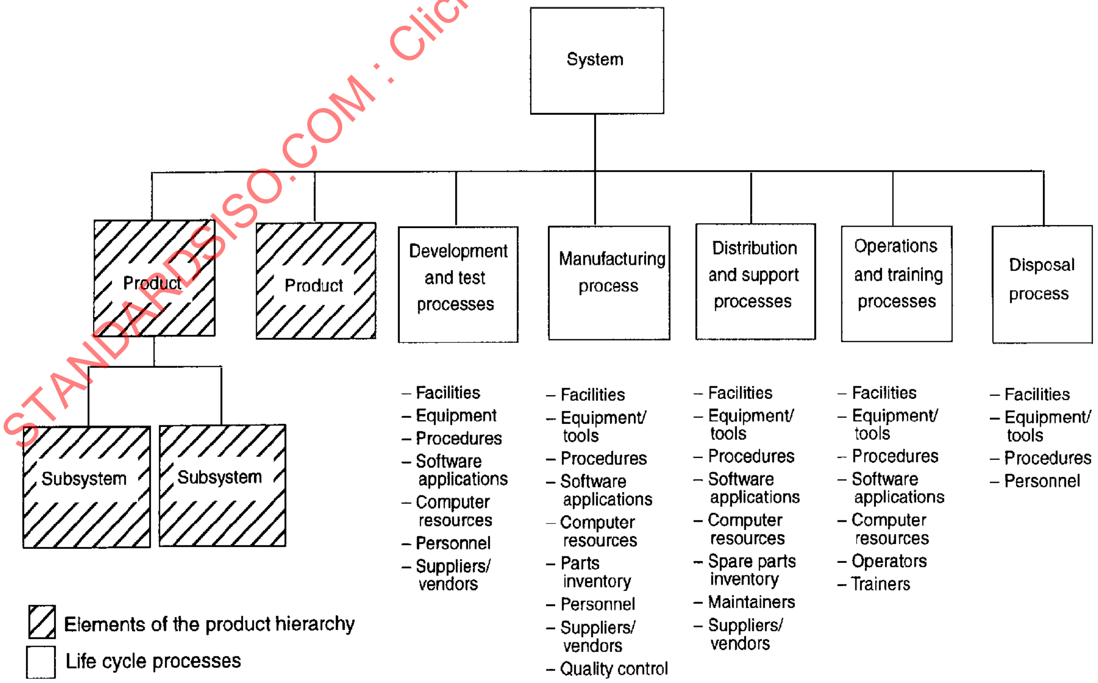


Figure 3—Life cycle process definition

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.

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.

1.4 Organization of this standard

- Clause 1 provides the scope, purpose, and organization of this standard.
- Clause 2 provides the normative 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 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 discusses some of the key differences between IEEE Std 1220 and ISO/IEC 15288:2002 [B3].
- Annex D provides bibliographic references.

2. Normative references

The following referenced documents are indispensable for the application of this standard. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

IEEE Std 610.12™-1990 (Reaff 2002), IEEE Standard Glossary of Software Engineering Terminology.[2]

[2] IEEE publications are available from the Institute of Electrical and Electronics Engineers, Inc., 445 Hoes Lane, Piscataway, NJ 08854, USA (http://standards.ieee.org/).

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.

3.1.1 acquirer: The stakeholder that acquires or procures a product or service from a supplier.

NOTE 1—Other terms commonly used for an *acquirer* are *buyer, customer,* and *purchaser.* The acquirer may at the same time be the owner, user, or operating organization.[3]

NOTE 2—*Customer* is a broader reference than *acquirer, operator,* or *user* and includes those roles as well as others.

NOTE 3—Adapted from ISO/IEC 15288:2002 [B3].

3.1.2 agreement: The mutual acknowledgement of terms and conditions under which a working relationship is conducted.

NOTE—See ISO/IEC 15288:2002 [B3].

3.1.3 allocation: The decision to assign a function or decision to hardware, software, or humans.

NOTE—Allocation may be made entirely to one of these three system element types or to some combination to be resolved upon further functional decomposition.

3.1.4 baseline: A specification or product that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development, and that can be changed only through formal change control procedures.

NOTE—See ISO/IEC 15288:2002 [B3].

3.1.5 constraint: A limitation or implied requirement that constrains the design solution or implementation of the SEP and is not changeable by the enterprise.

NOTE—A constraint is generally nonallocable.

3.1.6 customer: Organization or person that receives a product. Examples: Consumer, client, end user, retailer, beneficiary, and purchaser.

NOTE 1—A customer can be internal or external to the organization.

NOTE 2—See ISO 9000:2000 [B1].

NOTE 3—*Customer* is a broader reference than *acquirer, operator,* or *user* and includes those roles as well as others.

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.

NOTE—Cost goals should be treated with the same attention as any other performance parameter.

[3] Notes in text, tables, and figures are given for information only and do not contain requirements needed to implement the standard.

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

3.1.11 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.

3.1.12 end item: An entity (hardware equipment, software equipment, data, facilities, material, services, and/or techniques) identified with an element of the SBS.

3.1.13 enterprise: The organization that performs specified tasks.

NOTE—When a project that is in a nonconforming enterprise applies the standard directly, references to "enterprise" should be understood to mean the project.

3.1.14 facility: The physical means or equipment for facilitating the performance of an action, e.g. buildings, instruments, tools.

NOTE—See ISO/IEC 15288:2002 [B3].

3.1.15 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.

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

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

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

3.1.19 interface specification: The description of essential functional, performance, and design requirements and constraints at a common boundary between two or more system elements.

NOTE—This includes interfaces between humans and hardware or software, as well as interfaces between humans themselves.

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

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

3.1.22 life cycle model: A framework of processes and activities concerned with the life cycle, which also acts as a common reference for communication and understanding.

NOTE—See ISO/IEC 15288:2002 [B3].

3.1.23 measure of effectiveness (MOE): The metrics by which an acquirer will measure satisfaction with products produced by the technical effort.

3.1.24 measure of performance (MOP): An engineering performance measure that provides design requirements that are necessary to satisfy an MOE.

NOTE—There are generally several measures of performance for each MOE.

3.1.25 operator: An individual or an organization that contributes to the functionality of a system and draws on knowledge, skills, and procedures to contribute the function.

NOTE 1—The role of operator and the role of user may be vested, simultaneously or sequentially, in the same individual or organization.

NOTE 2—An individual operator combined with knowledge, skills, and procedures may be considered as an element of the system.

NOTE 3—*Customer* is a broader reference than *acquirer, operator,* or *user* and includes those roles as well as others.

NOTE 4—Adapted from ISO/IEC 15288:2002 [B3].

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

3.1.27 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).

3.1.28 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.

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

3.1.30 stage: A period within the life cycle of a system that relates to the state of the system description or the system itself.

NOTE 1—Stages relate to major progress and achievement milestones of the system through its life cycle.

NOTE 2—Stages may be overlapping.

NOTE 3—See ISO/IEC 15288:2002 [B3].

3.1.31 stakeholder: A party having a right, share, or claim in a system or in its possession of characteristics that meet that party's needs and expectations.

NOTE 1—*Stakeholder* is a broader reference than *customer, acquirer, operator,* or *user* and includes those roles as well as others.

NOTE 2—Adapted from ISO/IEC 15288:2002 [B3].

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

3.1.33 supplier: An organization or an individual that enters into an agreement with the acquirer for the supply of a product or service.

NOTE—See ISO/IEC 15288:2002 [B3].

3.1.34 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 operational needs and provides for the life cycle sustainment of the products.

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

3.1.36 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.

NOTE—The SBS can be used as a basis for cost-tracking and control.

3.1.37 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.

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

3.1.39 user: Individual who or group that benefits from a system during its utilization.

NOTE 1—The role of user and the role of operator may be vested, simultaneously or sequentially, in the same individual or organization.

NOTE 2—*Customer* is a broader reference than *acquirer, operator,* or *user* and includes those roles as well as others.

NOTE 3—Adapted from ISO/IEC 15288:2002 [B3].

3.2 Acronyms

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
SEMS	systems engineering master schedule
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 standard to develop a total system solution that is responsive to market opportunities, specified stakeholder requirements, enterprise objectives, and external constraints. Performance of the activities listed in items a) through f) are typically necessary to meet this goal.

To meet this goal, the enterprise should

a) 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) Apply the SEP for each level of system.

c) Control progress through the conduct of the following:

    1) Technical reviews following each level of development

    2) Risk management

    3) Data management

    4) Interface management

    5) Configuration management

    6) Performance-based progress measurement

d) Generate models and prototypes to support trade-off.

e) Generate an integrated data package, which ensures that the product can be produced, tested, delivered, operated, supported, and properly disposed of.

f) Capture the outputs from all technical activities in an integrated repository.

Specific requirements and recommendations are provided in other clauses of this standard.

4.1 Systems engineering process

The enterprise shall apply the normative provisions of 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 that 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 subprocesses of the SEP and shows how they iterate to produce a consistent set of requirements, functional arrangements, and design solutions. The subprocesses, tasks, and activities of the SEP 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 tailored and accomplished. Enterprise policies and procedures should address the following:

a) Application of the SEP throughout a project life cycle

b) Preparation and approval of a systems engineering management plan

c) Preparation and approval of a systems engineering master schedule (SEMS) and systems engineering detailed schedule

d) Preparation and approval of the integrated data package

e) Monitoring and reporting technical progress of the project

f) Preparation for, and conduct of, technical reviews

g) Contents and maintenance of an integrated repository

h) Continuous improvement of products and processes

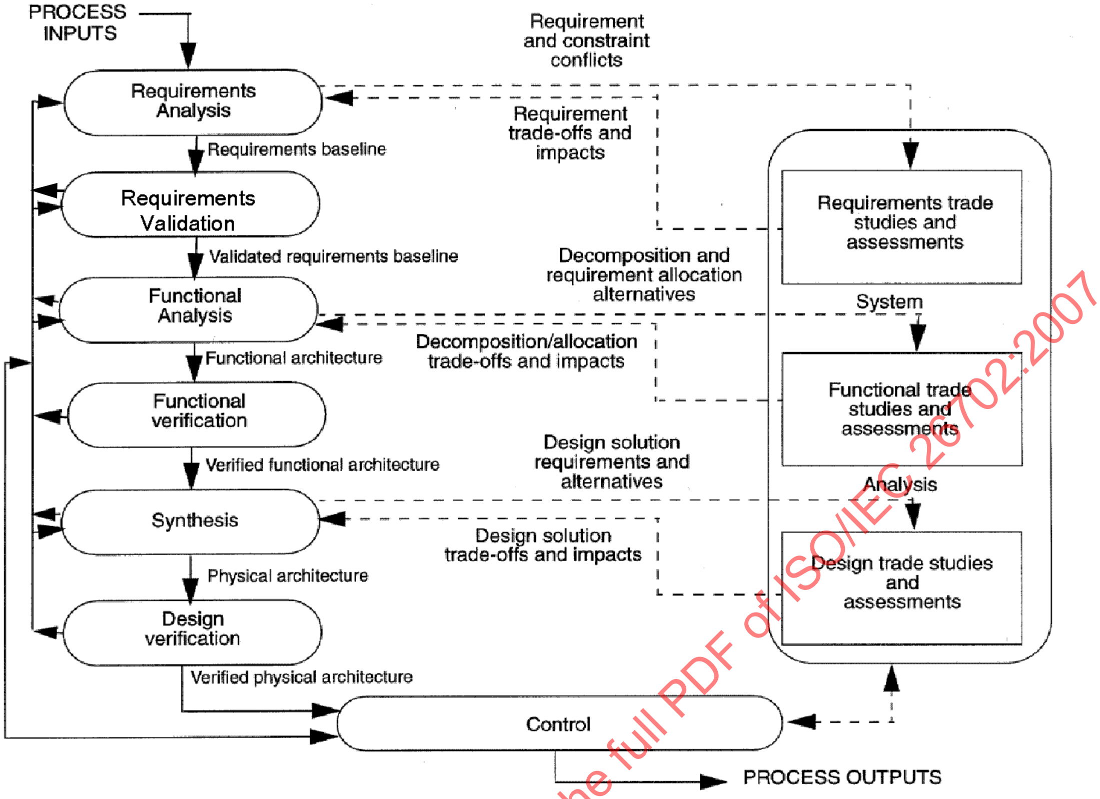

PROCESS INPUTS
Requirement and constraint conflicts
Requirements Analysis
Requirement trade-offs and impacts
Requirements baseline
Requirements Validation
Requirements trade studies and assessments
Validated requirements baseline
Decomposition and requirement allocation alternatives
Functional Analysis
System
Decomposition/allocation trade-offs and impacts
Functional architecture
Functional verification
Functional trade studies and assessments
Verified functional architecture
Design solution requirements and alternatives
Analysis
Synthesis
Design solution trade-offs and impacts
Physical architecture
Design trade studies and assessments
Design verification
Verified physical architecture
Control
PROCESS OUTPUTS

Figure 4—Systems engineering process (SEP)

4.3 Planning the technical effort

The project 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 project 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 describe 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 project 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 that addresses all factors associated with meeting system life cycle requirements. 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 capability and /or 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, configuration management, technical reviews, verifications, computer resources, manufacturing, maintenance, training, security, and human systems engineering.

4.4 Development strategies

The project 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 should be designed into the system architecture to enable the cost-effective sustainment of the system throughout its life cycle. This design attribute should be established early in the system development [when identifying stakeholder 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 project should determine and establish suitable models, simulations, or prototypes to support analysis and project decision making. Typical subjects of modeling, simulation, and prototyping include: requirements definition, analysis of the system architecture and design, and mitigation of identified risks. These simulations focus on ensuring 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 repository

The enterprise shall capture pertinent design data in a repository for the evolving integrated data package and to provide a shared resource for the exchange and reuse of technical information.

4.6.1 Data

Pertinent data includes: 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 project 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 project shall generate an integrated data package that documents architecture and design information for the support of life cycle processes. The integrated data package should include the types of engineering data defined in Table 1, as required by the project. The initial content of the integrated data package should 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.

4.7.1 Hardware

The integrated data package includes 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.

Table 1—Integrated data package content

An integrated data package includes, but is not limited to the following items:

A. Hardware

1) **Arrangement drawings**—document the relationship of the major subsystems or components of the system.

2) **Assembly drawings**—document the relationship of a combination of parts and subassemblies required to form the next higher indenture level of equipment or system.

3) **Connection drawings**—document the electrical connections of an installation or of its component devices or parts.

4) **Construction drawings**—document 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**—document complete end-item requirements for the subcomponent(s) delineated in the drawing.

7) **Elevation drawings**—document 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**—document general configuration and complete information necessary to install an item relative to its supporting structure or to associated items.

10) **Logic diagrams**—document, 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**—document complete design and functional engineering and product requirements of an item to facilitate production by tape control means.

12) **Piping diagrams**—document 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**—document a book-form drawing consisting of tabular data and instructions required to establish wiring connections within or between items.

14) **Schematic diagrams**—document, by means of graphical symbols, the electrical connections and functions of a specific circuit arrangement.

15) **Wiring and cable harness drawings**—document the path of a group of wires laced together in a specific configuration, so formed to simplify installation.

16) **Models, simulations, or design databases**—provide a physical, analytical, or digital representation of any of the items previously listed in items 1) through 15).

B. Software

1) **Software design documentation**—documents the software items architecture, design requirements, implementation logic, and data structures that provide a means of support.

2) **Software source code listings**—documents the actual source code instructions that represented the "as-built" implementation.

Table 1—Integrated data package content ***(continued)***

An integrated data package includes, but is not limited to the following items:
C. Human—documents cognitive, physical, and sensory characteristics of the humans who operate, maintain, and support the system throughout its life cycle that directly contribute to, or constrain, system performance and impact human-machine interfaces. 1) **Manpower, personnel, and training documentation**—documents the knowledge, skills, and abilities; training requirements; and availability of the humans who operate, maintain, and support the system throughout its life cycle. 2) **Workspace arrangement drawings**—document 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**—document 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**—document 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**—document the equipment/system design features, performance specifications, and training that reduces the potential for human or machine errors or failures that cause injury or death within the constraints of operational effectiveness, time, and cost throughout the equipment/system life cycle.
D. Life cycle processes **Process product design architecture**—documents the design architecture for the life cycle process products related to development (systems engineering and integration), manufacturing, verification, distribution, support, training, and disposal. Products include equipment, software, people, facilities, processes, and services integral to a specific life cycle process.

4.8 Specification tree

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

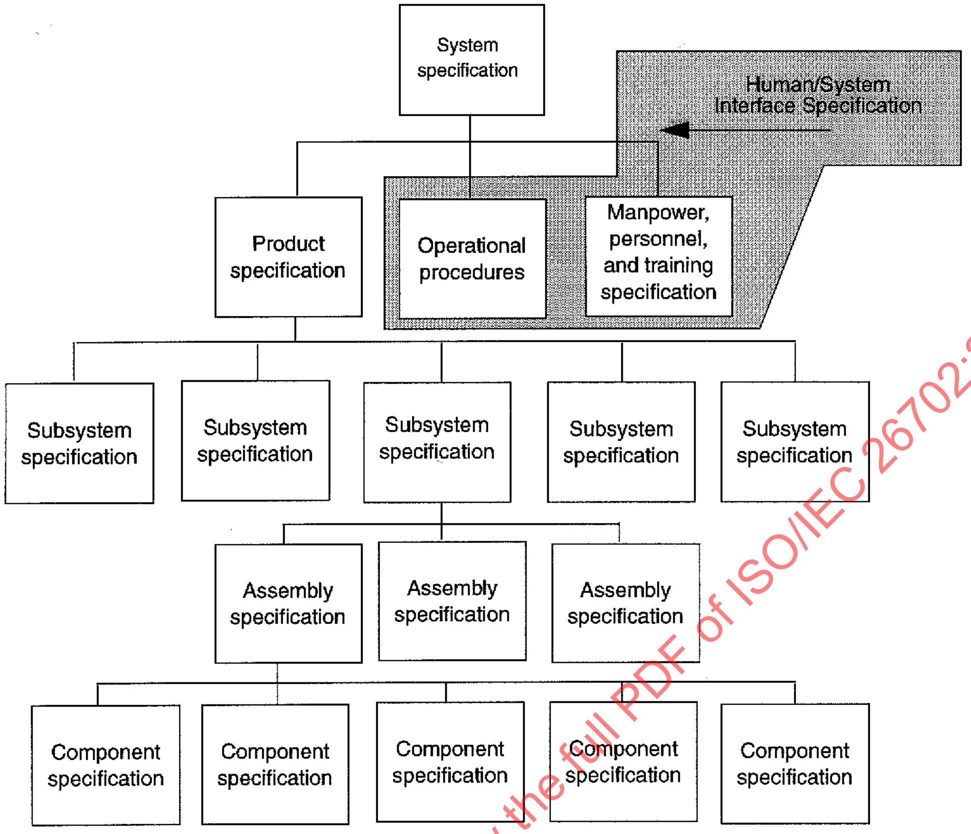


Figure 5—The specification tree

4.10 System breakdown structure

The project shall generate an SBS to depict the hierarchy of products and processes that comprise the system architecture. The SBS can be used by the project 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 6. 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.

4.11 Integration of the systems engineering effort

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

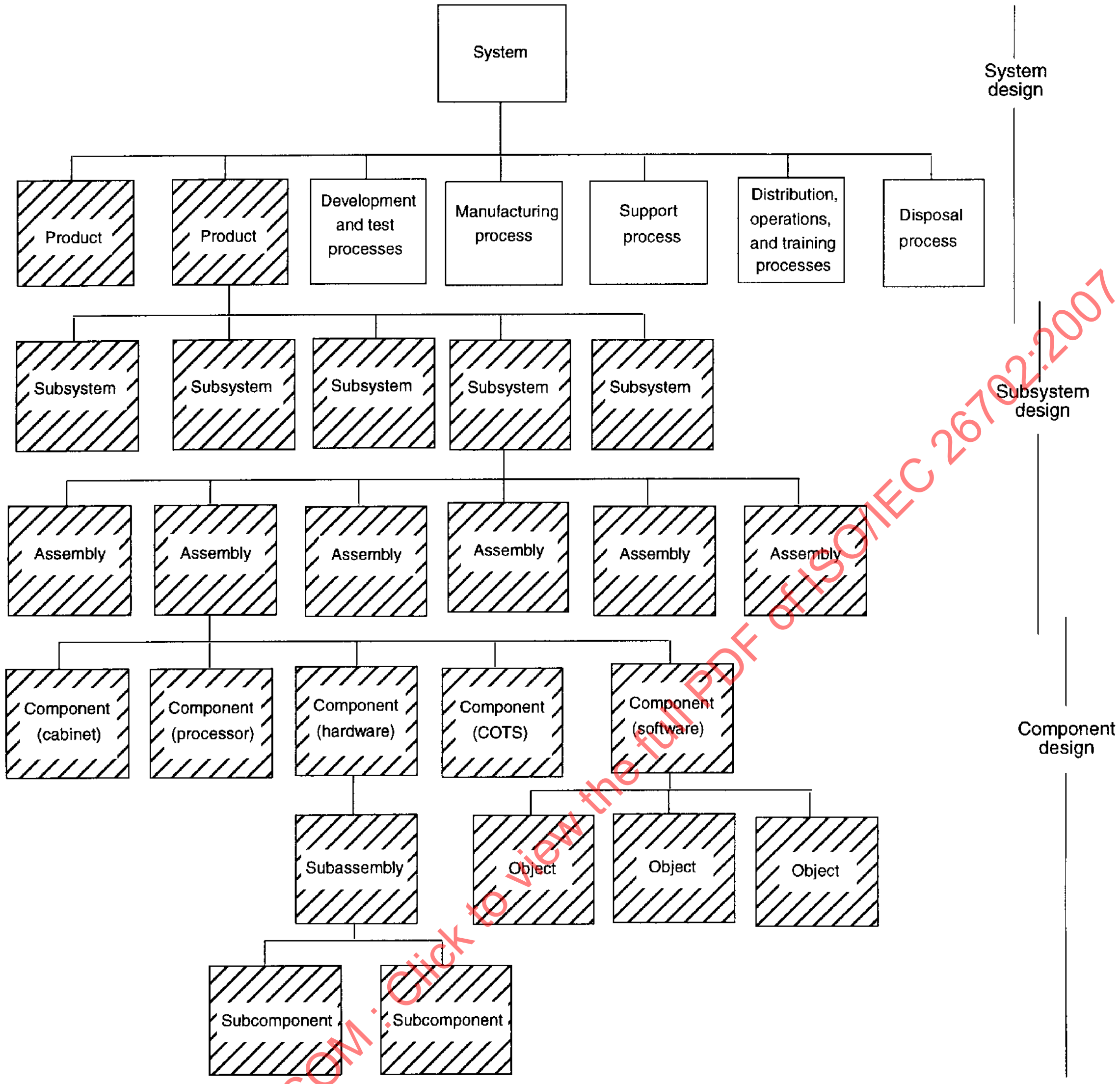


Figure 6—System breakdown structure

4.11.1 Concurrent engineering

The enterprise or project should 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 repository.

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 project should assign teams to specific elements of the SBS. Each team should prepare the required planning documents (engineering plan, master schedule, etc.) for the system 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 project shall conduct reviews, to include design reviews (e.g., system, subsystem, component, life cycle processes, test readiness, production approval) and audits (e.g., 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 should be conducted 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 should 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.

4.13 Quality management

The enterprise and project shall apply quality-management procedures for the development of products and life cycle processes.

4.14 Product and process improvement

The enterprise and project should establish and maintain product and process quality factors in order to continuously improve products and processes throughout the system life cycle in a manner consistent with enterprise objectives.

4.14.1 Re-engineering

The enterprise and project should explore ways to improve existing products and processes. The enterprise and project 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.

NOTE—Re-engineering is the process of improving a system after production through modification to correct a design deficiency or to make an incremental improvement.

4.14.2 Self-assessment

The enterprise should 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.

4.14.3 Lessons learned

The enterprise should capture the lessons learned on each project and incorporate them into enterprise training, 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 project 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. Applying the SEP during each level of system development (system, subsystem, and component) adds value (additional detail) to the products defined in the prior application of the process. Application of the SEP is also useful during fabrication, assembly, integration, and test (FAIT), production, and consumer support stages to resolve reported problems, and for evolving products 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) FAIT

Stages of operations

a) Production

b) Support

These stages are depicted in Figure 7.

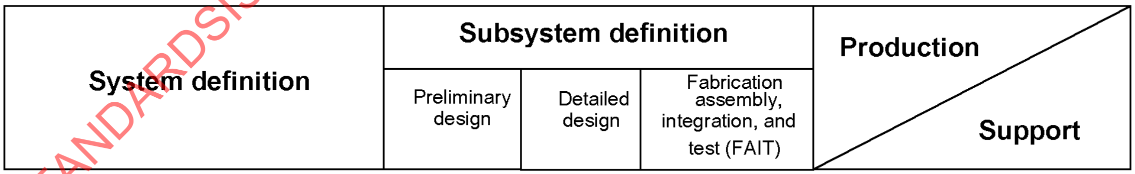


Figure 7—Typical system life cycle

The project 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 project 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 project 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 project should 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 project applies 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

A precedented system is one 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. For precedented system developments, with incrementally improved or evolutionary growth products, the project should refine established system definitions and product-line strategies to satisfy the market opportunity or order.

An unprecedented system is one for which design examples do not exist so that the design architecture alternatives are unconstrained by previous system descriptions. For unprecedented systems where the concept is not already defined, the project should create and evaluate candidate alternative concepts that would satisfy a market opportunity or 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 project 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 project should establish the necessary engineering and technical plans for the system life cycle, which include the following:

a) The activity accomplishment criteria for determining system definition progress assessment

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 project should identify the subsystems of each product and define the design and functional interface requirements among the subsystems and their corresponding performance requirements and design

Table 2—System definition

Establish system definition
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
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
Complete specifications
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
Establish baselines
a) Establish system baseline
b) Establish preliminary subsystem “design-to” baselines
Complete technical reviews
a) Complete alternative concept review
b) Complete system definition review

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 project should identify the interface requirements between humans and products or subsystems. These interface requirements include performance, workloads, design constraints, and usability.

5.1.1.5 Life cycle quality factors

The project should 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 that ensures that quality factors traceability is maintained.

5.1.1.6 Revised project and engineering plans

The project should 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 project should 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 project should 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) should 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 project 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 project 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 project should 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 project should 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 project should prepare a specification for the personnel required to operate, maintain, and support the system throughout its life cycle. Analysis should 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 project should 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 project should 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 repository 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 project should 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 project should 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 project 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 the following:

a) Product and subsystem allocations being reasonable and providing a sound system concept

b) The capability of the concept to satisfy stakeholder 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 stakeholder requirements and meets public expectations

5.1.4.2 System definition review

A system definition review should be completed by the project 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 project should execute the preliminary design stage to initiate subsystem design and create subsystem-level specifications and design-to baselines to guide component development. The project applies 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 subsystem-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 project should 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.

5.2.1.1 Assemblies and assembly interfaces

The project should 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 project should 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 project should mitigate subsystem-level risks that were assessed to be critical to subsystem development during system definition and should assess and mitigate assembly risks associated with each subsystem. For

Table 3—Subsystem definition—preliminary design

Establish preliminary subsystem definitions	
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	
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	
a)	Update system baselines
b)	Establish design-to baselines
c)	Establish preliminary “build-to” baselines
Complete technical reviews	
a)	Complete subsystem preliminary design reviews
b)	Complete system preliminary design review

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 project 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 project should 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 project should 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 project 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 project should 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 project 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 project 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 project 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 should be satisfied during component development. The project should identify which assembly interface requirements provide constraints to component design and which should have changes managed through interface control working groups.

5.2.2.4 Preliminary component specifications

The project 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 project should 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 project should update the specifications for the personnel required to operate, maintain, and support the system throughout its life cycle. Analysis should 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 project 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 project should update and control all changes to the system baseline.

5.2.3.2 Design-to baseline

The project 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 repository, 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 project 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 project 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 are not normally held as part of a project management review.

5.2.4.1 Subsystem reviews

Subsystem reviews should be completed by the project 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 project 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 project should 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 project applies 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 project applies 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.

Table 4—Subsystem definition—detailed design

Establish detailed subsystem definitions 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 FAIT
Complete specifications 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 a) Update system and design-to baselines b) Establish build-to baselines
Complete technical reviews a) Complete component detailed design reviews b) Complete subsystem detailed design reviews c) Complete system detailed design review

5.3.1.1 Component definition

The project 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 that ensures that requirements traceability is maintained in both directions.

5.3.1.2 Component risks

The project 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 project 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 project should 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 project 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 that ensures that quality-factors traceability is maintained.

5.3.1.5 Integrated data package

The project 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 repository 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 project 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 project 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 project 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 project 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 project should 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 project should update the specifications for the personnel required to operate, maintain, and support the system throughout its life cycle. Analysis should 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 project 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 project 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 project 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 repository, 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 project 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 project 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.

NOTE—Effectiveness criteria is the measure of value used to determine the success or failure of a design solution.

5.3.4.2 Subsystem reviews

Subsystem-level reviews should be completed by the project 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 project 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 stage

The project applies 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.

5.4.1 System integration and test

The project should 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 project executes the test activities to verify that the system meets system requirements. The project completes this by first testing the components and then conducting tests at each level up to, and including, the total system. The project should 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 project 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 project should analyze the deficiency to determine the cause of the problem and apply the SEP to resolve the problem. The project should then retest the product to ensure operational effectiveness, usability,

Table 5—Subsystem definition—FAIT

Conduct system integration and test	
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
Complete technical reviews	
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

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

5.4.3 Project and engineering plans

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

5.4.5 Configuration baselines

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

5.4.6 Technical reviews

The project 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 project 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 confirms 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) should be completed by the project 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 should be completed by the project 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 project 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 support stages

The project applies 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 project also applies the SEP during 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	Support
Produce system products 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 a) Complete component physical configuration audits b) Complete subsystem physical configuration audits c) Complete system physical configuration audits	Provide operator and user services a) Provide services b) Provide aftermarket products Complete system evolution a) Evolve design to 1) Make an incremental change 2) Resolve product deficiencies 3) Exceed competitive products

Figure 8—Production and support

5.5.1 System products

The project 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 project 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 project should 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 project 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, projects 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 project to assure the system elements conform to the technical documentation that defines the build-to baseline.

5.5.3 Support

Once products are in the field, the project should continue to support operators and users with needed services and aftermarket products and to maintain supplier relationships.

5.5.4 System evolution

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

5.5.4.2 Specifications

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

5.5.4.3 Configuration baselines

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

5.6 Simultaneous engineering of life cycle processes

The project should 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 project 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 the definition of the product (system product, subsystem, assembly, or component subassembly) to be supported, the project should schedule applicable downstream life cycle processes to be available at the time needed to enable the process to be supportive of that product. 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.

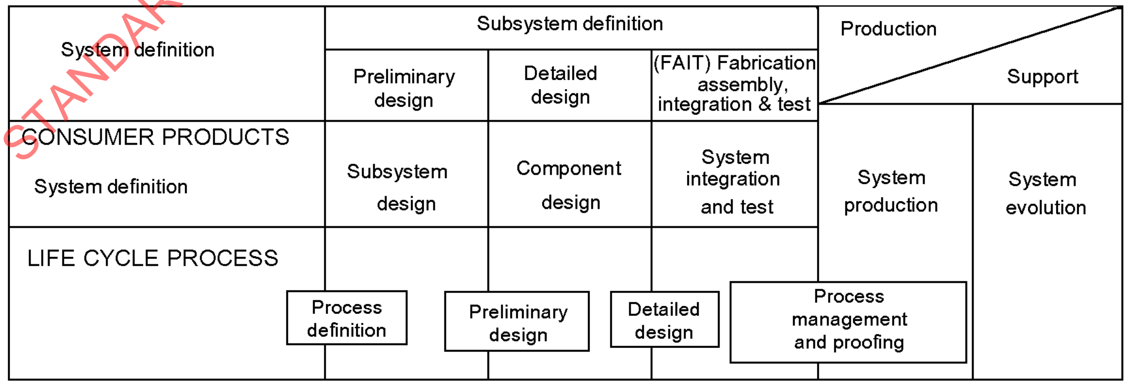


Figure 9—Simultaneous engineering of life cycle processes

5.6.1 Life cycle process development

The project 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 project 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 project 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

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. Enterprise policies and procedures describe the general application and conduct of subprocess tasks throughout a project life cycle. The project 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 and in accordance with enterprise policies and procedures.

6.1 Requirements analysis

The project shall perform the tasks of requirements analysis for the purpose of establishing what the system will be capable of accomplishing; how well system products are to perform in quantitative, measurable terms; the environments in which system products operate; the requirements of the human/system interfaces; the physical/aesthetic characteristics; and constraints that affect design solutions. The market needs, requirements, and constraints are derived from stakeholder 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 to be solved. The tasks associated with requirements analysis are identified in Figure 10. The project 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 project 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 stakeholder expectations

The project defines and quantifies the stakeholder expectations for the system. Stakeholder expectations may come from marketing, an acquirer’s order, a recognized market opportunity, direct communications from users, or the requirements from a higher-level system. Stakeholder expectations include the following:

a) What the stakeholder wants the system [product(s), life cycle processes, and desired quality factors] to accomplish (functional requirements)

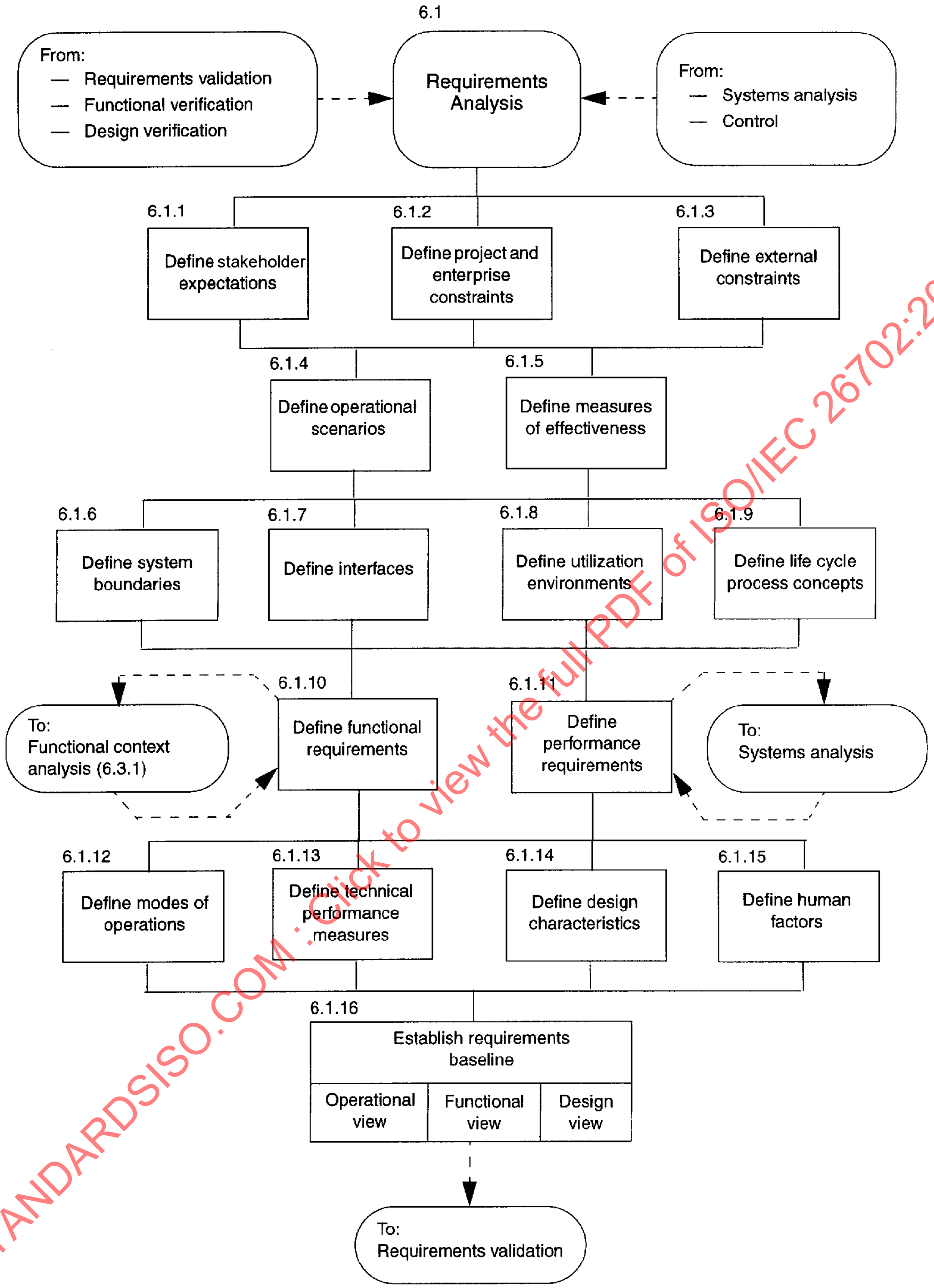

6.1
Requirements Analysis
From:
— Requirements validation
— Functional verification
— Design verification
From:
— Systems analysis
— Control
6.1.1
Define stakeholder expectations
6.1.2
Define project and enterprise constraints
6.1.3
Define external constraints
6.1.4
Define operational scenarios
6.1.5
Define measures of effectiveness
6.1.6
Define system boundaries
6.1.7
Define interfaces
6.1.8
Define utilization environments
6.1.9
Define life cycle process concepts
To:
Functional context analysis (6.3.1)
6.1.10
Define functional requirements
6.1.11
Define performance requirements
To:
Systems analysis
6.1.12
Define modes of operations
6.1.13
Define technical performance measures
6.1.14
Define design characteristics
6.1.15
Define human factors
6.1.16
Establish requirements baseline
Operational view
Functional view
Design view
To:
Requirements validation

Figure 10—Requirements analysis

b) How well each function is to 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 stakeholder expectations is balanced with an analysis of the 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 the following:

a) Approved specifications and baselines developed from prior applications of the SEP

b) 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

g) Reuse and commercial-off-the-shelf (COTS)

6.1.2.2 Enterprise constraints

Enterprise constraints may include the following:

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

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 project identifies and defines external constraints that impact design solutions or the implementation of SEP activities. These constraints include the following:

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 project identifies and defines the operational scenarios that define the range of the anticipated uses of system product(s). For each operational scenario, the project 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

The project defines system effectiveness measures that reflect overall stakeholder 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 project defines the following:

a) Which system elements are under design control of the project and which fall outside their control

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 project 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 project defines the utilization environments for each of the operational scenarios. All environmental factors, natural or induced, that may affect system performance are identified and defined. Factors that 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 environmental factors are defined for possible locations and conditions where the system may be operated. Effects on hardware, software, and humans are assessed for impact on system performance and life cycle processes.

6.1.9 Define life cycle process concepts

The project 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 project 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 project 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 project 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 project 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 project 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 project performs functional context analysis (see 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 are further decomposed during functional decomposition (see 6.3.2) to provide a basis for identifying and assessing design alternatives. All requirements of a system typically involve a functional and performance aspect, and it is viewing system requirements as having both functional and performance aspects that ensures that the requirements are complete, consistent, and verifiable.

6.1.11 Define performance requirements

The project 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 project 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

The project identifies the technical performance measures (TPMs), which are key indicators of system performance. Selection of TPMs are usually 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 project identifies and defines required design characteristics (e.g., color, texture, size, anthropomorphic limitations, weight, and buoyancy) for the system products under development. The project identifies which design characteristics are constraints and which can be changed based on trade-off analyses.

6.1.15 Define human factors

The project 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 project 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 project. 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 may 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 time lines 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 that limit design solutions and/or development procedures
    8) Design capabilities and capacities for planned evolutionary growth

6.2 Requirements validation

The project shall perform the tasks of requirements validation. During requirements validation activity

a) The requirements baseline that was established during requirements analysis is evaluated to ensure that it represents identified stakeholder expectations and project, enterprise, and external constraints.
b) The requirements baseline is assessed 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 repository and is an input to functional analysis. The tasks associated with requirements validation are identified in Figure 11.

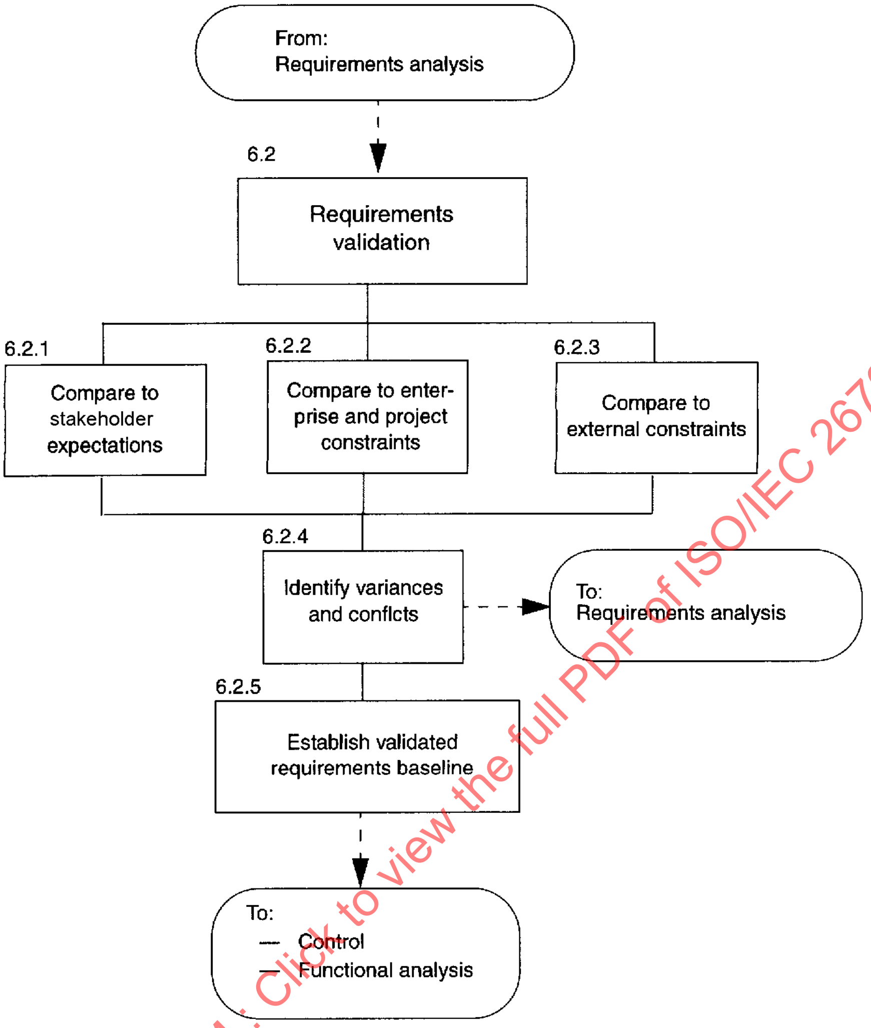

From:
Requirements analysis
6.2
Requirements validation
6.2.1
Compare to stakeholder expectations
6.2.2
Compare to enterprise and project constraints
6.2.3
Compare to external constraints
6.2.4
Identify variances and conflcts
To:
Requirements analysis
6.2.5
Establish validated requirements baseline
To:
— Control
— Functional analysis

Figure 11—Requirements validation

6.2.1 Compare to stakeholder expectations

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

6.2.2 Compare to enterprise and project constraints

The project 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 project 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 project identifies and defines variances and conflicts that arise out of the validation tasks 6.2.1 through 6.2.3. Each variance or conflict is 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 (see 6.3) and documented in the integrated repository.

6.3 Functional analysis

The project shall perform the tasks of functional analysis to accomplish the following two related objectives:

a) To describe the problem defined by requirements analysis in clearer detail

b) To decompose the system functions to lower-level functions that should 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 (see 6.5) set the criteria that guide the definition of product and subsystem solutions. The tasks associated with functional analysis are identified in Figure 12.

6.3.1 Functional context analysis

The project 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 nonstressful 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 project identifies these interfaces and defines their functional interactions, such as start and end states or inputs and outputs.

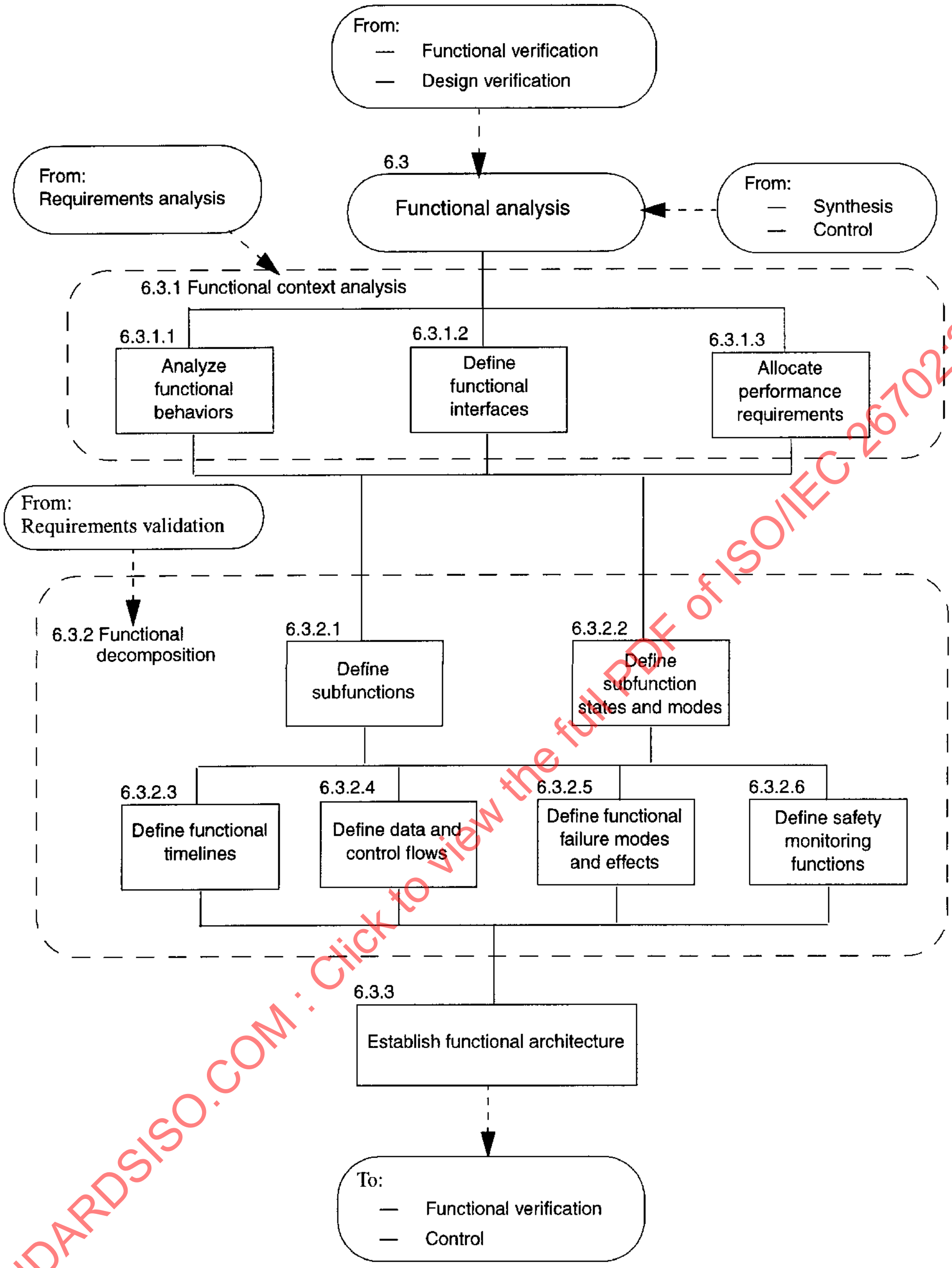

From:
— Functional verification
— Design verification
6.3
Functional analysis
From:
Requirements analysis
From:
— Synthesis
— Control
6.3.1 Functional context analysis
6.3.1.1
Analyze functional behaviors
6.3.1.2
Define functional interfaces
6.3.1.3
Allocate performance requirements
From:
Requirements validation
6.3.2 Functional decomposition
6.3.2.1
Define subfunctions
6.3.2.2
Define subfunction states and modes
6.3.2.3
Define functional timelines
6.3.2.4
Define data and control flows
6.3.2.5
Define functional failure modes and effects
6.3.2.6
Define safety monitoring functions
6.3.3
Establish functional architecture
To:
— Functional verification
— Control

Figure 12—Functional analysis process

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 project documents the allocation of system performance requirements to functions to provide traceability and to facilitate later changes.

6.3.2 Functional decomposition

The project decomposes the system into subfunctions with the intent of defining the following:

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 (see 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 time lines, 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 project 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 project selects the best functional architecture and documents it in the integrated repository.

6.3.2.2 Define subfunction states and modes

The project 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 time line

The project analyzes sequences of subfunctions and their behaviors to identify and define a functional time line 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 time line.

6.3.2.4 Define data and control flows

The project 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 project 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 project 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 project establishes the functional architecture, appropriate to the level of development, to define the allocation of performance requirements from which design solutions should be determined via synthesis (see 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 project shall conduct the tasks of functional verification 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 associated with functional verification are identified in Figure 13.

6.4.1 Define verification procedures

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

6.4.2 Conduct verification evaluation

The project 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 project 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 project 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 project 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 project identifies variances and conflicts resulting from verification evaluation activities of task 6.4.2. When incompleteness is shown, functional analysis tasks (see 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 (see 6.3) be repeated to eliminate nonrequired functions and/or performance requirements. For the latter, requirements analysis (see 6.1) and validation (see 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 repository. This verified functional architecture is used in synthesis to generate design solutions to satisfy stakeholder expectations and meet public acceptance as defined by the validated requirements baseline.

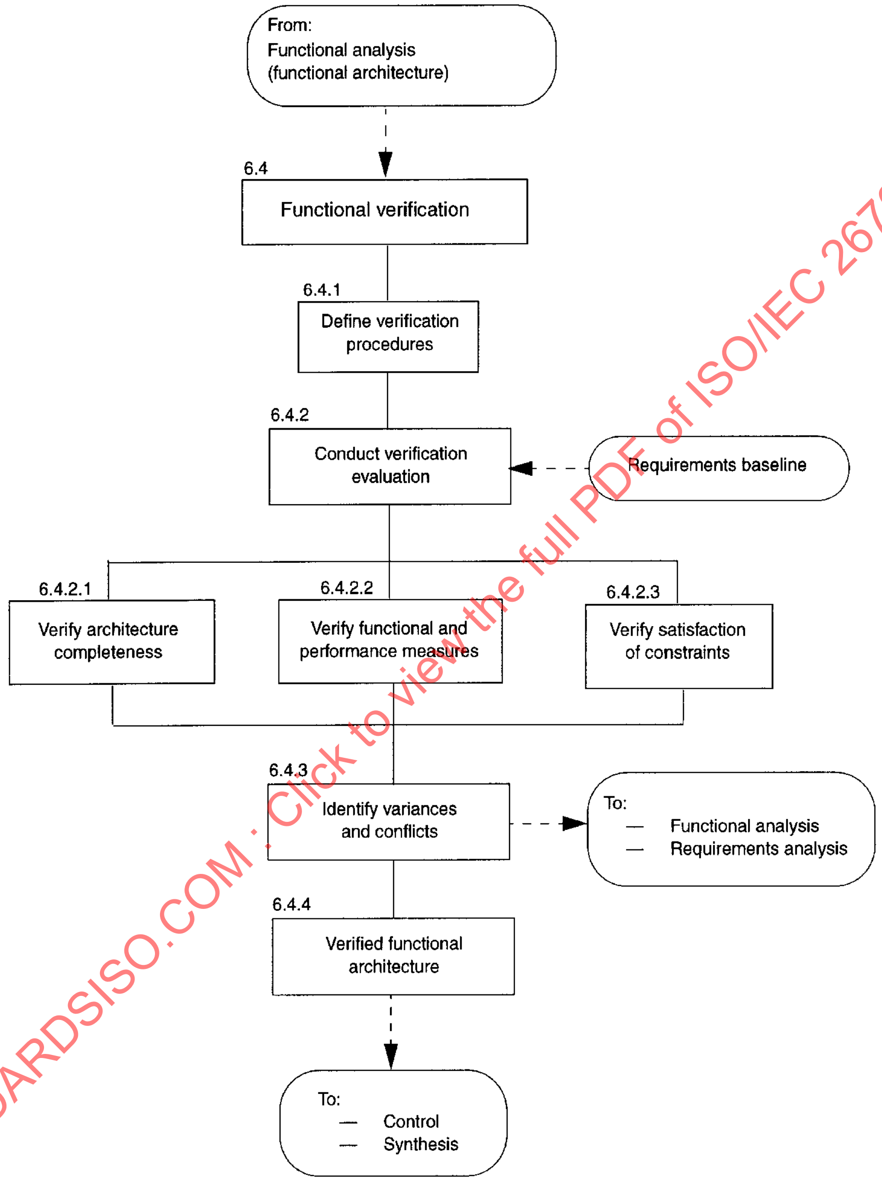

From:
Functional analysis
(functional architecture)
6.4
Functional verification
6.4.1
Define verification procedures
6.4.2
Conduct verification evaluation
Requirements baseline
6.4.2.1
Verify architecture completeness
6.4.2.2
Verify functional and performance measures
6.4.2.3
Verify satisfaction of constraints
6.4.3
Identify variances and conflicts
To:
— Functional analysis
— Requirements analysis
6.4.4
Verified functional architecture
To:
— Control
— Synthesis

Figure 13—Functional verification

6.5 Synthesis

The project shall perform the tasks of 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 (see 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 associated with synthesis are identified in Figure 14.

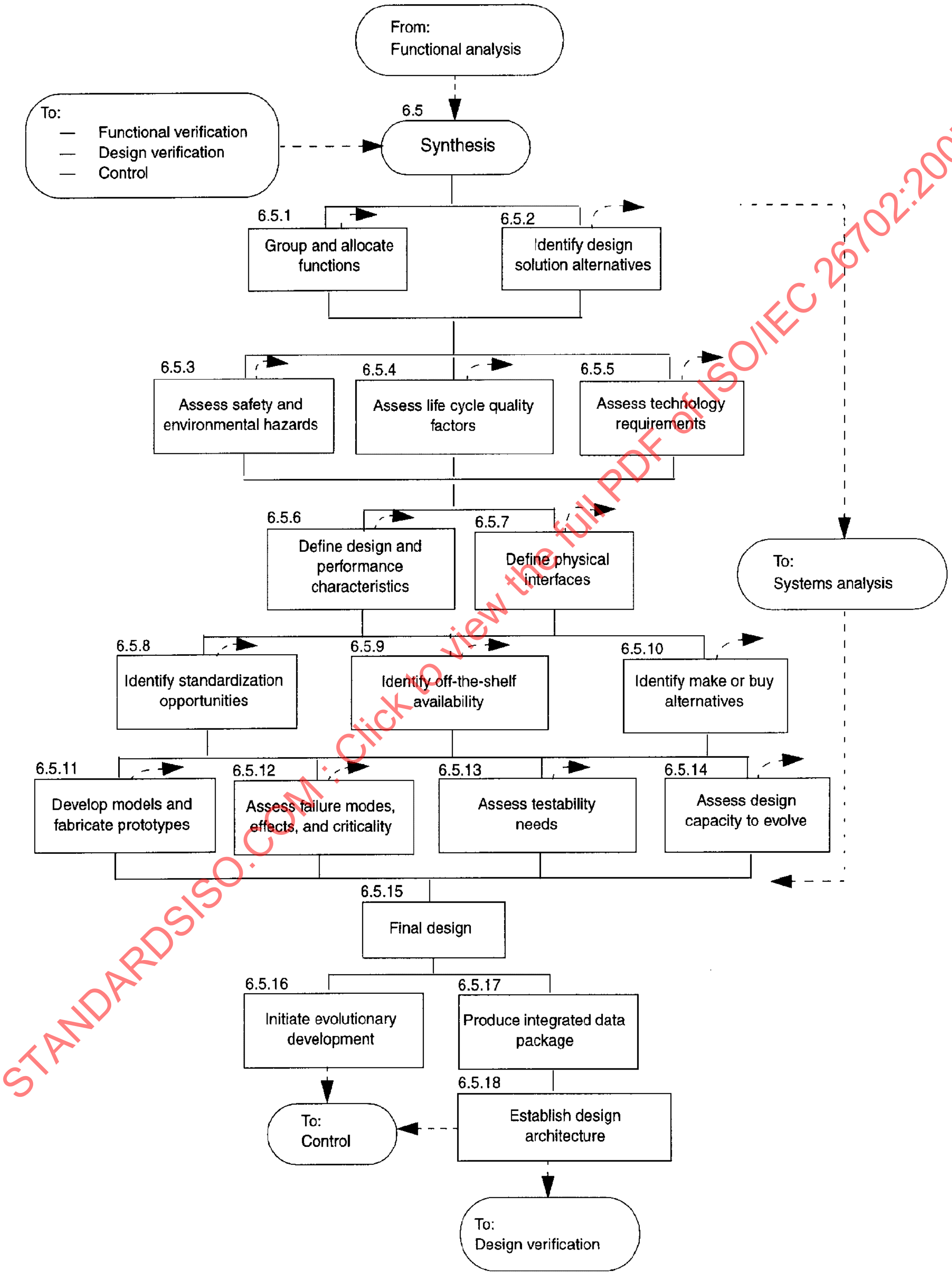

From:
Functional analysis
To:
— Functional verification
— Design verification
— Control
6.5
Synthesis
6.5.1
Group and allocate functions
6.5.2
Identify design solution alternatives
6.5.3
Assess safety and environmental hazards
6.5.4
Assess life cycle quality factors
6.5.5
Assess technology requirements
6.5.6
Define design and performance characteristics
6.5.7
Define physical interfaces
To:
Systems analysis
6.5.8
Identify standardization opportunities
6.5.9
Identify off-the-shelf availability
6.5.10
Identify make or buy alternatives
6.5.11
Develop models and fabricate prototypes
6.5.12
Assess failure modes, effects, and criticality
6.5.13
Assess testability needs
6.5.14
Assess design capacity to evolve
6.5.15
Final design
6.5.16
Initiate evolutionary development
6.5.17
Produce integrated data package
6.5.18
Establish design architecture
To:
Control
To:
Design verification

Figure 14—Synthesis process

6.5.1 Group and allocate functions

The project 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 (see 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 project 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 project 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 project 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 project assesses whether associated life cycle process needs, requirements, and constraints of the processes are identified and defined.

6.5.5 Assess technology requirements

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

6.5.11 Develop models and prototypes

The project develops models and/or prototypes to assist in the following:

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 repository. 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 project 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 project 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 project 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, users, 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 project 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 project 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 project 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 project 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 project 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 project 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 documented in the integrated repository, 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 (see 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 project shall perform the tasks of 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.

b) The design architecture satisfies the validated requirements baseline.

The tasks associated with design verification are identified in Figure 15.

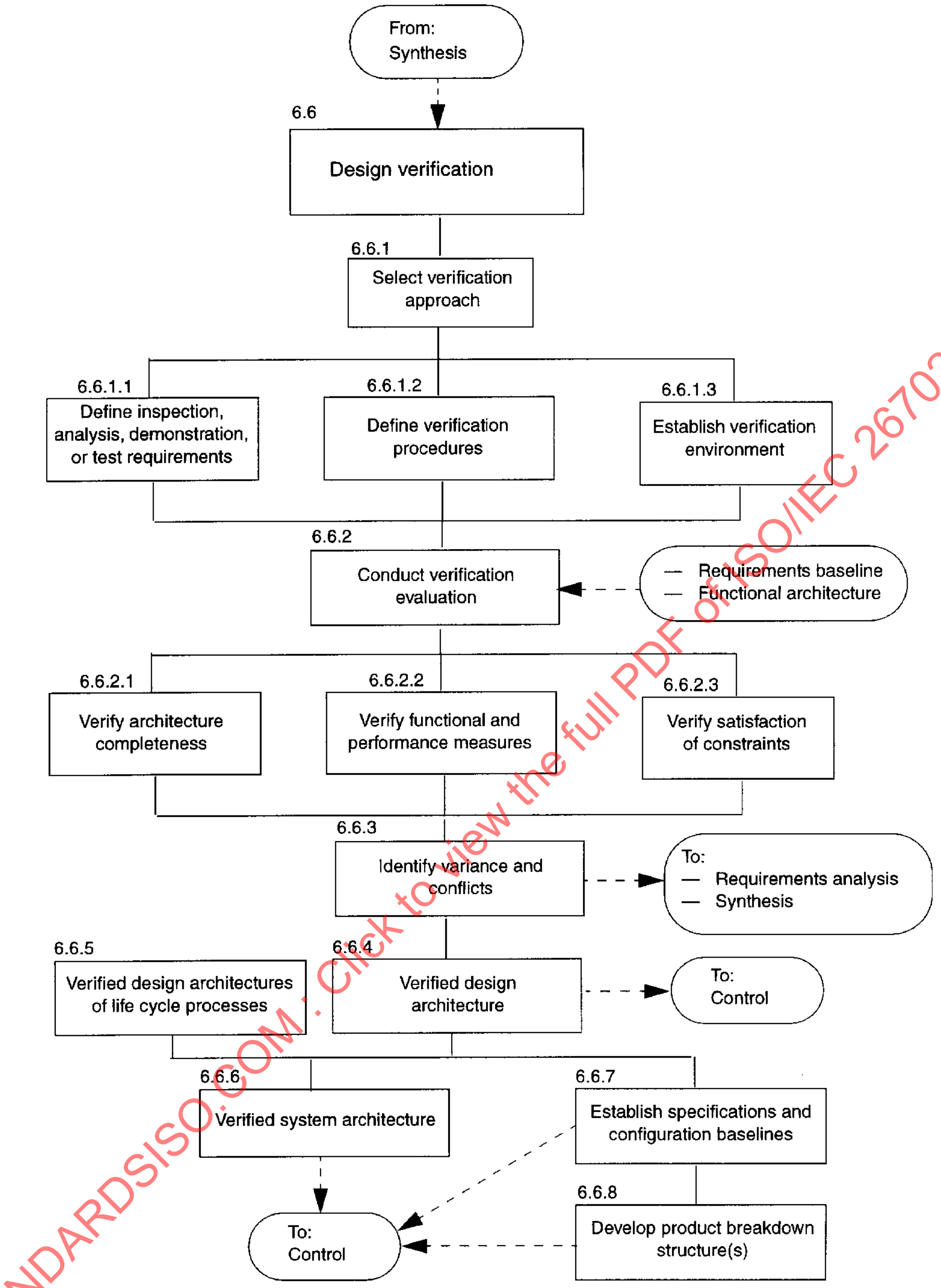

From:
Synthesis
6.6
Design verification
6.6.1
Select verification approach
6.6.1.1
Define inspection, analysis, demonstration, or test requirements
6.6.1.2
Define verification procedures
6.6.1.3
Establish verification environment
6.6.2
Conduct verification evaluation
— Requirements baseline
— Functional architecture
6.6.2.1
Verify architecture completeness
6.6.2.2
Verify functional and performance measures
6.6.2.3
Verify satisfaction of constraints
6.6.3
Identify variance and conflicts
To:
— Requirements analysis
— Synthesis
6.6.5
Verified design architectures of life cycle processes
6.6.4
Verified design architecture
To:
Control
6.6.6
Verified system architecture
6.6.7
Establish specifications and configuration baselines
6.6.8
Develop product breakdown structure(s)
To:
Control

Figure 15—Design verification

6.6.1 Select verification approach

The project 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 project selects the appropriate verification method [inspection, analysis (including mock-ups or simulations), demonstration, or test] for evaluating whether functional and performance requirements, and design characteristics 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 project 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 project defines procedures for each verification method selected, identifies the purpose and objectives of each verification procedure, identifies the pretest and post-test 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 project 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 project 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 project verifies that

a) Design elements descriptions are traceable to requirements of the functional architecture (upward traceability).

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 project 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 project verifies that the

a) Evaluation results from activities, as defined by 6.6.1, satisfy the constraints, including interfaces, of the functional architecture.

b) Constraints of the established design architecture are traceable to the validated requirements baseline.

6.6.3 Identify variances and conflicts

The project identifies variances and conflicts resulting from verifying activities. When variances show incompleteness, synthesis tasks (see 6.5) or functional analyses tasks (see 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 no required 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 situation indicates that synthesis (see 6.5) be repeated to eliminate nonrequired functions and/or performance requirements. For the latter variance or conflict, requirements analysis through functional verification (see 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 repository. 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 project completes requirements analysis, functional analysis, and synthesis tasks to identify, define, and design architectures for life cycle processes. The project 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 project 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 project 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 documented in the integrated repository and used in the next application of the SEP. The design solution for the next level of development may satisfy these specifications.

6.6.8 Develop system breakdown structure (SBS)

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

6.7 Systems analysis

The project shall perform the tasks of systems analysis 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 associated with systems analysis are identified in Figure 16. Even if a trade-off analysis is not done, an overall assessment of the system effectiveness should be completed.

6.7.1 Assess requirement conflicts

The project 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 project 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 project 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 project 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 the following:

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.

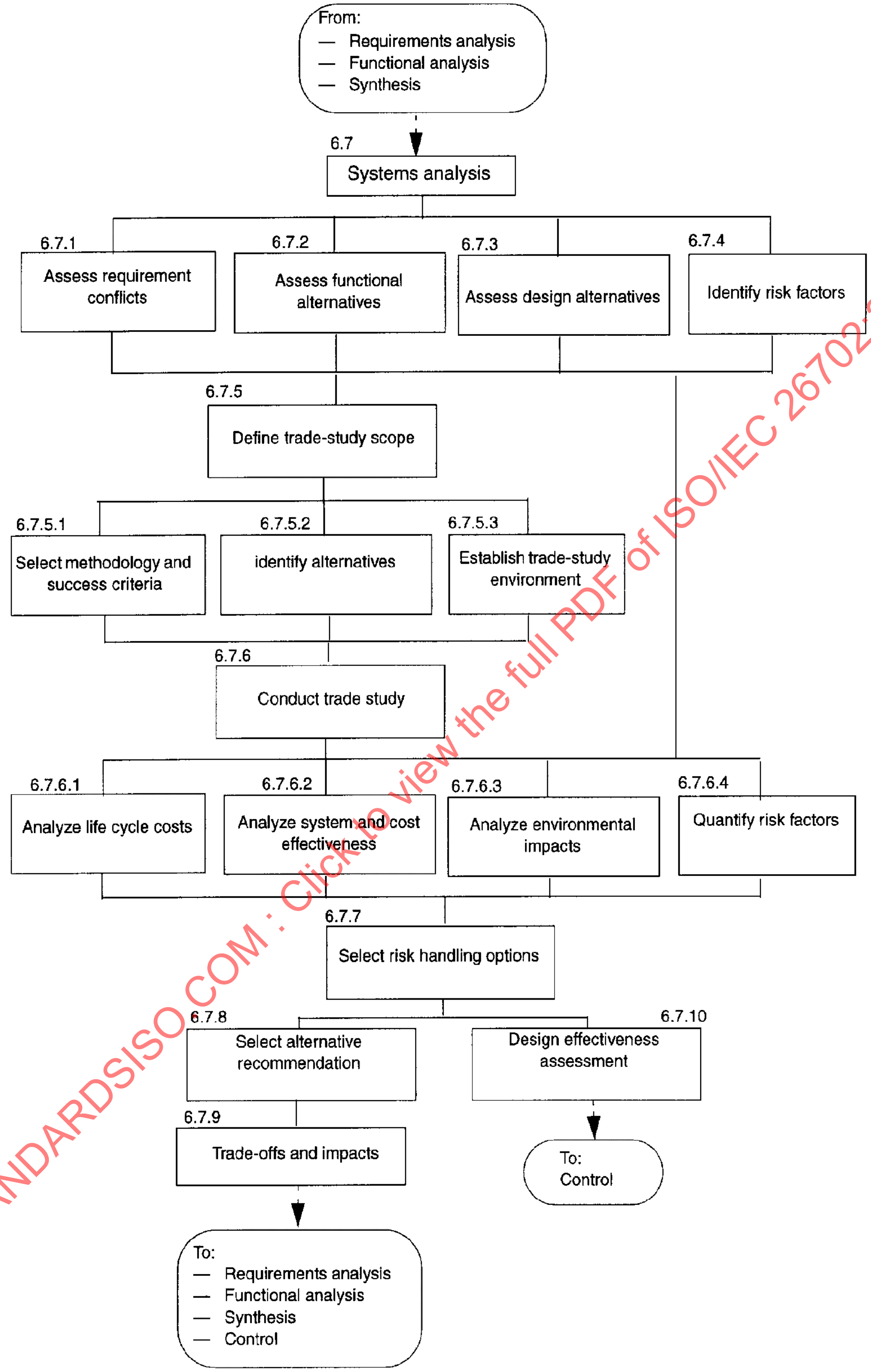


Figure 16—Systems analysis process

6.7.5 Define trade-off analysis scope

The project 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 judgmental 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 acquirer;

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 stakeholder 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 project 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 project 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 project 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 project establishes metrics for each criterion that characterizes how well various alternatives satisfy the criterion. In addition, the project 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 project completes tasks 6.7.6.1 through 6.7.6.4, to the degree appropriate, to complete trade-off analyses for the following:

a) Requirements analysis to both resolve conflicts with and satisfy stakeholder/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 project analyzes the costs to the project and to the acquirer 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 project 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 project identifies safety and environmental impacts associated with system implementation. Applicable environmental laws and regulations should be identified, and the project should ensure that these are complied with by any alternative solution. The project 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 that present a known hazard to the environment are to be avoided to the extent feasible. Where not feasible, provisions may 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 project 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 project 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 project 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 project utilizes the results of trade-off analyses and risk-reduction planning information to recommend a preferred alternative to the decision maker. The project 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 project 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 repository.

6.7.10 Design effectiveness assessment

The project 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 repository and briefed at appropriate technical and project reviews.

6.8 Control

The project shall perform the tasks of control for the purpose of managing and documenting the activities of the SEP. The tasks associated with 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 project. The control tasks provide the following:

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 support

d) Information for decision makers at technical and project reviews

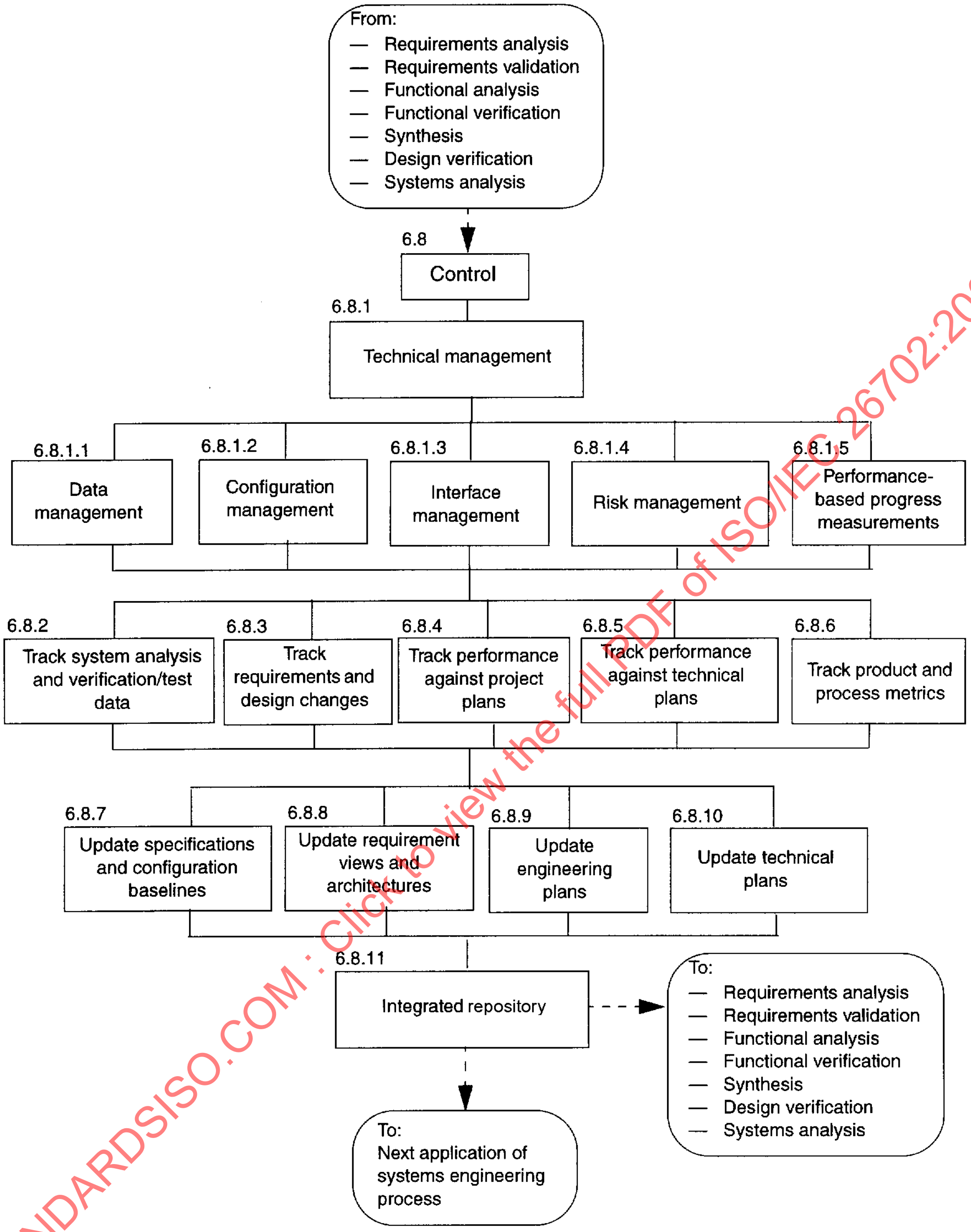

From:
— Requirements analysis
— Requirements validation
— Functional analysis
— Functional verification
— Synthesis
— Design verification
— Systems analysis
6.8
Control
6.8.1
Technical management
6.8.1.1
Data management
6.8.1.2
Configuration management
6.8.1.3
Interface management
6.8.1.4
Risk management
6.8.1.5
Performance-based progress measurements
6.8.2
Track system analysis and verification/test data
6.8.3
Track requirements and design changes
6.8.4
Track performance against project plans
6.8.5
Track performance against technical plans
6.8.6
Track product and process metrics
6.8.7
Update specifications and configuration baselines
6.8.8
Update requirement views and architectures
6.8.9
Update engineering plans
6.8.10
Update technical plans
6.8.11
Integrated repository
To:
— Requirements analysis
— Requirements validation
— Functional analysis
— Functional verification
— Synthesis
— Design verification
— Systems analysis
To:
Next application of systems engineering process

Figure 17—Control process

6.8.1 Technical management

The project manages the tasks and activities of the SEP to control data generated, configuration of the design solutions, interfaces, risks, and technical progress. The project needs to maintain the correct staffing, facilities, equipment, and tools; manage costs and schedules; replan development activities, as required; coordinate technical interactions with stakeholders; 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 project 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 repositories 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 project 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 project plans and implements the functions of the following:

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 project establishes and maintains configuration control boards to process, review, and approve engineering changes to end items.

6.8.1.3 Interface management

The project 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 project establishes and maintains interface working groups to process, review, and make recommendations on approval of interface changes.

6.8.1.4 Risk management

The project should conduct risk management to systematically control the uncertainty in the project's ability to meet cost, schedule, and performance objectives. The project 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 that 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 project 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 project 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 project 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 project 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 project 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 should be requested in advance and should be undertaken only when approved.

6.8.5 Track progress against engineering plans

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

a) Determine technical areas requiring project management attention.

b) Determine the degree of stakeholder satisfaction and public acceptance.

c) Provide cost and schedule estimates for new products and provide faster response to stakeholders.

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

1) Establishment of a quality system and achievement of efficient use of resources

2) Overall system quality and productivity evaluation

3) Comparison to planned goals and targets

4) Early detection of problems

5) Benchmarking of the SEP

6.8.7 Update specifications and configuration baselines

The project 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 project updates requirements views and the functional, design, and system architectures to reflect changes brought about by an acquirer, 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 project should update the engineering plans to reflect changes brought about by an acquirer, 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 project should update the technical plans to reflect changes brought about by an acquirer, 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 repository

The project establishes and maintains a repository of all pertinent data and information from tasks 6.8.1 through 6.8.10. This repository contains all significant information used and generated by the SEP and describes the current state of the system development and its evaluation. An electronic medium is preferred. As a shared resource, the repository needs to be accurate, unambiguous, secure, survivable, easily accessible by authorized users, and complete.

Annex A

(informative)

The role of systems engineering within an enterprise

A.1 The systems engineering process

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.

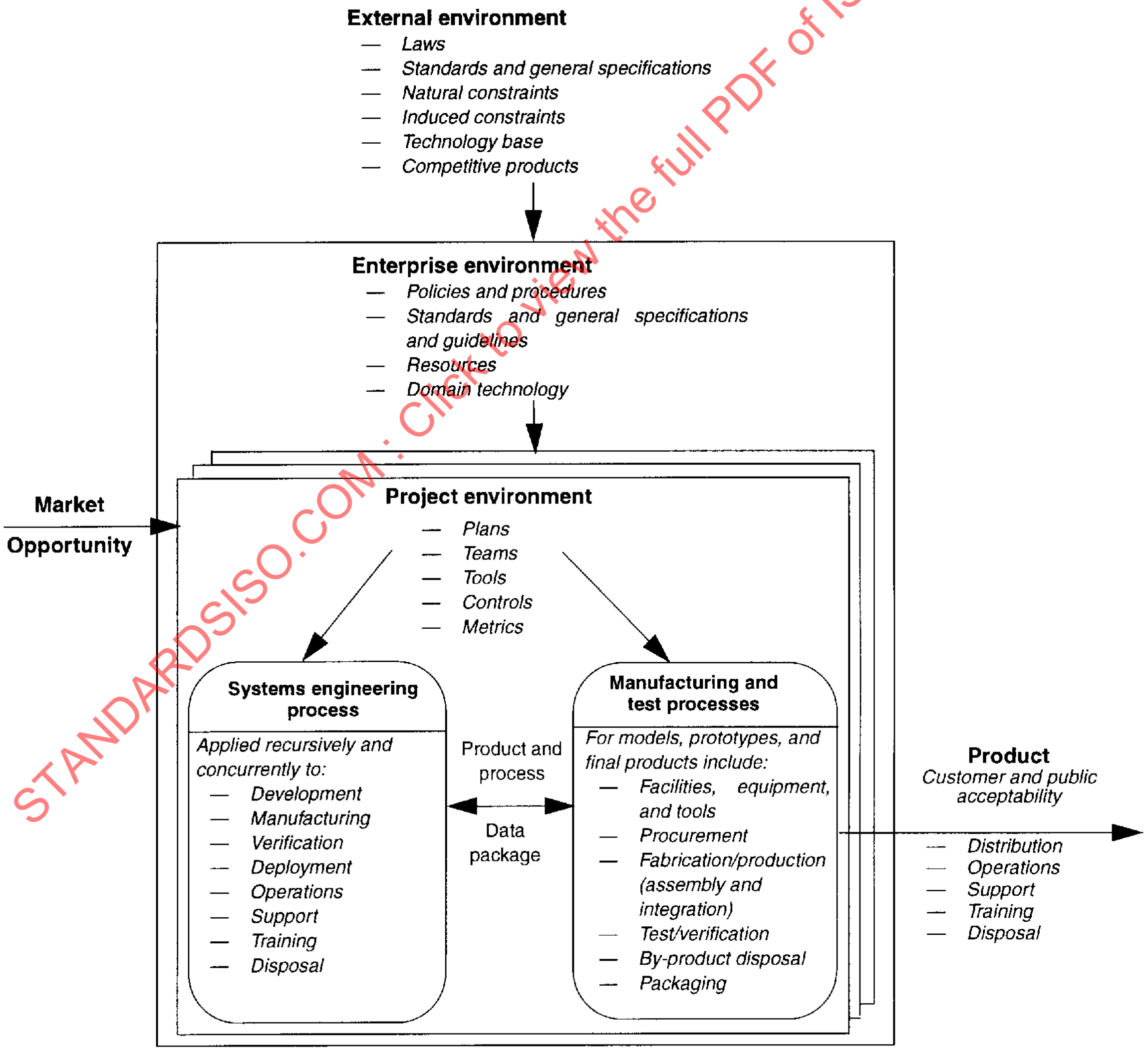


Figure A.1—Systems engineering environment within an enterprise