IEEE Std 1233-1996

IEEE Guide for Developing System Requirements Specifications

Sponsor

Software Engineering Committee of the IEEE Computer Society

Approved 17 April 1996

IEEE Standards Board

Abstract: Guidance for the development of the set of requirements, System Requirements Specification (SyRS), that will satisfy an expressed need is provided. Developing a SyRS includes the identification, organization, presentation, and modification of the requirements. Also addressed are the conditions for incorporating operational concepts, design constraints, and design configuration requirements into the specification. This guide also covers the necessary characteristics and qualities of individual requirements and the set of all requirements.
Keywords: requirement, system, system requirements specifications

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

ISBN 1-55937-716-X

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

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

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

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

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

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

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 all 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.

Introduction

(This introduction is not a part of IEEE Std 1233-1996, IEEE Guide for Developing System Requirements Specifications.)

The purpose of this guide is to provide guidance for capturing system requirements. This guide serves the analyst by providing a clear definition for identifying well-formed requirements and ways of organizing them.

This guide should be used to help the analyst capture requirements at the beginning of the system requirements phase. It should be used to clarify what constitutes a good requirement and provide an understanding of where to look to identify different requirement sources.

This guide was prepared by a working group chartered by the Software Engineering Committee of the Computer Society of IEEE. This guide represents a consensus of individual working-group participants with broad expertise in systems engineering and requirements analysis.

The following individuals contributed to the writing of this guide by attending two or more working sessions, by substantial written commentary, or both.

At the time this guide was completed, the Working Group had the following membership:

Louis E. Miller, *Chair*
William N. Sabor, *Secretary*

Bakul Banerjee	P. Michael Guba	Jim Longbucco
David Byrch	James R. Hughes	Donald F. Parsons
Kim A. Cady	Joe Iaquinto	Eric Peterson
Larry Diehr	Marybeth A. Jupina	John Sheckler
Charles A. Droz	Thomas M. Kurihara	Jess Thompson
Larry C. Forrest	Richard C. Lee	Eva D. Williams

Other contributors include:

Geoff Cozens	Kristin Dittmann	Virginia Nuckolls
Paul Davis	Christof Ebert	Anne O'Neill
	Don McCash	

The following persons were on the balloting committee:

H. Ronald Berlack	Patrick J. Griffin	Indradeb P. Pal
Mark Bilger	David A. Gustavson	Joseph A. Palermo
William J. Boll	John Harauz	Stephen R. Schach
Fletcher Buckley	Derek J. Hatley	Norman Schneidewind
Edward R. Byrne	William Hefley	Wolf A. Schnoege
Francois Coallier	Umesh P. Hiriyannaiah	Gregory D. Schumacher
Christopher Cooke	Jody Howard	Carl S. Seddio
Geoff Cozens	Eiichi Kaneko	David M. Seifert
Alan M. Davis	Jerry Kickenson	Richard S. Sky
Robert E. Dwyer	Janet Kintner	Alfred R. Sorkowitz
Sherman Eagles	Thomas M. Kurihara	Robert N. Sulgrove
Leo G. Egan	Renee Lamb	Tanehiro Tatsuta
Caroline L. Evans	John B. Lane	Richard H. Thayer
Richard L. Evans	Boniface Lau	George D. Tice
John W. Fendrich	J. Dennis Lawrence	Leonard L. Tripp
Peter Fillery	Ben Livson	Tom Vaiskunas
Larry Forrest	Harold Mains	Thomas E. Vollman
Eugene P. Friedman	Roger Martin	Ronald L. Wade
Eitan Froumine	James W. Mc Clean	Dolores Wallace
Yair Gershkovitch	Sue McGrath	William M. Walsh
Adel N. Ghannam	Louis E. Miller	Paul R. Work
Julio Gonzalez Sanz	Dennis E. Nickle	Janusz Zalewski

When the IEEE Standards Board approved this standard on 17 April 1996, it had the following membership:

Contents

IEEE Guide for Developing System Requirements Specifications

1. Overview

1.1 Scope

This standard provides guidance for the development of a set of requirements that, when realized, will satisfy an expressed need. In this standard the set of requirements will be called the System Requirements Specification (SyRS). Developing an SyRS includes the identification, organization, presentation, and modification of the requirements. This guide addresses conditions for incorporating operational concepts, design constraints, and design configuration requirements into the specification. This guide also addresses the necessary characteristics and qualities of individual requirements and the set of all requirements.

This guide does not specify industry wide system specification standards nor state mandatory system requirements specification. This guide is written under the premise that the current state of the art of system development does not warrant or support a formal standards document.

2. References

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

ASTM E 623-89, Standard Guide for Developing Functional Requirements for Computerized Systems.[1]

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

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

IEEE Std 730-1989, IEEE Standard for Software Quality Assurance Plans (ANSI).

IEEE Std 828-1990, IEEE Standard for Software Configuration Management Plans (ANSI).

[1]ASTM publications are available from the American Society for Testing and Materials, 100 Barr Harbor Drive, West Conshohocken, PA 19428-2959, USA.
[2]IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, USA.

IEEE Std 830-1993, IEEE Guide for Software Requirements Specifications (ANSI).

IEEE Std 1074-1995, IEEE Standard for Developing Software Life Cycle Processes (ANSI).

IEEE Std 1220-1994, IEEE Trial-Use Standard for Application and Management of the Systems Engineering Process.

ISO 9000: 1994, Quality management and quality assurance standards: Guidelines for selection and uses.[3]

ISO 9126: 1991, Information tech. software system evaluation, Quality characteristics and guidelines for their use.

MIL-STD-490/DI-CMAN-80008A/DI-CMAN-80008, Military Standard Specification Practices.[4]

MIL-STD 498, Military Standard Defense System Software Development.

3. Definitions

The definitions listed below establish meaning in the context of this guide. Terms not defined in this standard are included in IEEE Std 610.12-1990.[5]

3.1 analyst: A member of the technical community (such as a systems engineer or business analyst, developing the system requirements) who is skilled and trained to define problems and to analyze, develop, and express algorithms.

3.2 annotation: Further documentation accompanying a requirement such as background information and/or descriptive material.

3.3 baseline: A specification or system that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development and can be changed only through formal change control procedures (IEEE 610.12-1990).

3.4 constraint: A statement that expresses measurable bounds for element or function of the system. That is, a constraint is a factor that is imposed on the solution by force or compulsion and may limit or modify the design changes.

3.5 customer(s): The entity or entities for whom the requirements are to be satisfied in the system being defined and developed. This can be an end-user of the completed system, an organization within the same company as the developing organization (e.g., System Management), a company or entity external to the developing company, or some combination of all of these. This is the entity to whom the system developer must provide proof that the system developed satisfies the system requirements specified.

3.6 derived requirement: A requirement deduced or inferred from the collection and organization of requirements into a particular system configuration and solution.

3.7 element: A component of a system; may include equipment, a computer program, or a human.

[3]ISO publications are available from the ISO Central Secretariat, Case Postale 56, 1 rue de Varembé, CH-1211, Genève 20, Switzerland/Suisse. ISO publications are also available in the United States from the Sales Department, American National Standards Institute, 11 West 42nd Street, 13th Floor, New York, NY 10036, USA.

[4]MIL publications are available from Customer Service, Defense Printing Service, 700 Robbins Ave., Bldg. 4D, Philadelphia, PA 19111-5094.

[5]Information about references can be found in clause 2.

3.8 end user: The person or persons who will ultimately be using the system for its intended purpose.

3.9 environment: The circumstances, objects, and conditions that will influence the completed system; they include political, market, cultural, organizational, and physical influences as well as standards and policies that govern what the system must do or how it must do it.

3.10 function: A task, action, or activity that must be accomplished to achieve a desired outcome.

3.11 model: A representation of a real world process, device, or concept.

3.12 prototype: An experimental model, either functional or non functional, of the system or part of the system. A prototype is used to get feedback from users for improving and specifying a complex human interface, for feasibility studies, or identifying requirements.

3.13 raw requirement: An environmental or customer requirement that has not been analyzed and formulated as a well-formed requirement.

3.14 representation: A likeness, picture, drawing, block diagram, description, or symbol that logically portrays a physical, operational, or conceptual image or situation.

3.15 requirement: (a) A condition or capability needed by a user to solve a problem or achieve an objective. (b) A condition or capability that must be met or possessed by a system or system component to satisfy a contract, a standard, specification, or other formally imposed documents. (c) A documented representation of a condition or capability as in (a) or (b). (IEEE Std 610.12-1990)

3.16 system: An interdependent group of people, objects, and procedures constituted to achieve defined objectives or some operational role by performing specified functions. A complete system includes all of the associated equipment, facilities, material, computer programs, firmware, technical documentation, services, and personnel required for operations and support to the degree necessary for self-sufficient use in its intended environment.

3.17 System Requirements Specification (SyRS): The System Requirements Specification consists of a structured collection of information that embodies the requirements of the system.

3.18 testability: The degree to which a requirement is stated in terms that permit establishment of test criteria and performance of tests to determine whether those criteria have been met. (IEEE Std 610.12-1990)

3.19 traceability: The degree to which a relationship can be established between two or more products of the development process, especially products having a predecessor-successor or master-subordinate relationship to one another; for example, the degree to which the requirements and design of a given system element match. (IEEE 610.12-1990)

3.20 validation: The process of evaluating a system or component during or at the end of the development process to determine whether a system or component satisfies specified requirements. (IEEE Std 610.12-1990)

3.21 verification: The process of evaluating a system or component to determine whether the system of a given development phase satisfy the conditions imposed at the start of that phase. (IEEE Std 610.12-1990)

3.22 well-formed requirement: A statement of system functionality (a capability) that can be validated, must be met or possessed by a system to solve a customer problem or to achieve a customer objective, and is qualified by measurable conditions and bounded by constraints.

4. System requirements specification

A System Requirements Specification (SyRS) has traditionally been viewed as a document that communicates the requirements of the customer to the technical community who will specify and build the system. The collection of requirements that constitutes the specification and its representation acts as the bridge between the two groups and must be understandable by both the customer and the technical community. One of the most difficult tasks in the creation of a system is that of communicating to all of the subgroups within both groups, especially in one document. This type of communication generally requires different formalisms and languages.

4.1 Definition

The SyRS presents the results of the definition of need, the operational concept, and the system analysis tasks. As such, it is a description of what the system's customers expect it to do for them, the system's expected environment, the system's usage profile, its performance parameters, and its expected quality and effectiveness. Thus it presents the conclusions of the SyRS Development process described in clause 5.

This guide suggests a distinction between this structured collection of information and the way in which it is presented to its various audiences.The presentation of the SyRS should take a form that is appropriate for its intended use.This can be a paper document, models, prototypes, other non-paper document representations, or any combination. All of these representations can be derived from this one SyRS to meet the needs of a specific audience. However, care should be taken to ensure that each of these presentations be traceable to a common source of system requirements information. The audience should be made aware that this structured collection of information remains the one definitive source for resolving ambiguities in the particular presentation chosen.

This guide makes a clear distinction between the system requirements (what the system must do) contained in the SyRS and process requirements (how to construct the system) that should be contained in contract documents like a Statement of Work.

4.2 Properties

The collection of requirements should have the following properties:

a) Unique Set: Each requirement should be stated only once.
b) Normalized: Requirements should not overlap, that is, they shall not refer to other requirements or the capabilities of other requirements.
c) Linked Set: Explicit relationships should be defined among individual requirements to show how the requirements are related to form a complete system.
d) Complete: An SyRS should include all the requirements identified by the customer, as well as those needed for the definition of the system.
e) Consistent: SyRS content should be consistent and non contradictory in the level of detail, style of requirement statements, and in the presentation of material.
f) Bounded: The boundaries, scope, and context for the set of requirements should be identified.
g) Modifiable: The SyRS should be modifiable. Clarity and non overlapping requirements contribute to this.
h) Configurable: Versions should be maintained versions time and across instances of the SyRS.
i) Granular: This should be the level of abstraction for the system being defined.

4.3 Purpose

The purpose of the SyRS is to provide a "black-box" description of what the system should do, in terms of the system's interactions or interfaces with its external environment. The SyRS should completely describe all inputs, outputs, and required relationships between inputs and outputs. The SyRS organizes and communicates requirements to the customer and technical community.

4.3.1 Organizing requirements

The purpose of the SyRS can best be achieved by organizing the system requirements into conceptual categories. In practice, it is difficult to identify and separate requirements from other aspects of the customer's perception of the system that are often included in documents that define "requirements." Often, traditional user procedures or user or technical community assumptions about the implementation cloud the fundamental statement of need. The analyst should capture and state the fundamental needs of the customer and the technical community, properly form requirements, and organize or group them into meaningful categories.

While organizing the unstructured user's statements into a structured set of requirements, the analyst should identify technical requirements without being diverted into stating an implementation approach. To be distracted into implementation issues before a clear understanding of the requirements is achieved may lead to both an inadequate statement of requirements and a faulty implementation. Discerning between technical requirements and technical implementations is a constant challenge to the analyst.

The description of the system should be stated in operational and logistical terms. Issues addressed include the system's desired operational capabilities, physical characteristics, performance parameters and expected values, interfaces with its environment, interactions with its environment, documentation requirements, reliability requirements, logistical requirements, and personnel requirements.

These requirements should be communicated in a structured manner to ensure that the customer and technical community are able to do the following:

a) Identify requirements that are derived from other requirements
b) Organize requirements of different levels of detail into their appropriate levels
c) Verify the completeness of the set of requirements
d) Identify inconsistencies among requirements
e) Clearly identify the capabilities, conditions, and constraints for each requirement
f) Develop a common understanding with the customer of the purpose and objectives of the set of requirements
g) Identify requirements that will complete the SyRS

It is important that structure be added to the set of requirements by the analysts, and that representations of the SyRS communicate the requirements in a structured manner. Clause 6 provides guidelines for explicitly defining the requirements.

4.3.2 Communicating to two audiences

The SyRS has two primary audiences and essentially serves to document an agreement between the customer and the technical community.

4.3.2.1 Customer

The customer is a collective term that may include customer of the proposed system, the funding agency, the acceptor who will sign-off delivery, and managers who will be responsible for overseeing the implementation, operation, and maintenance of the system.

All customers will have vested interests and concerns that should be resolved in the SyRS. In addition, some customers may not understand the process of establishing requirements or the process of creating a system. Although competent in their areas of responsibility and in the application for which the system is being defined, they generally may not be familiar with the vocabulary and representation techniques that are often used to specify requirements. Since one of the primary goals of System Requirements Analysis is to ensure that the SyRS is understood, it will be necessary to provide the customers with a representation of the SyRS in a language that the customer understands and that is complete, concise, and clear.

4.3.2.2 Technical community

The SyRS should also communicate the customer's requirements to the technical community. The technical community includes analysts, estimators, designers, quality assurance officers, certifiers, developers, engineers, integrators, testers, maintainers, and manufacturers. For this audience the representation of the SyRS should be technically precise and presented in a formalism from which they can design, build, and test the required system.

4.4 Intended use

The recommended uses of the SyRS vary as the development cycle progresses are as follows:.

a) During Systems Design, requirements are allocated to subsystems, hardware, software, operations, and other major components of the system.
b) The SyRS is utilized in constructing the resulting system. The SyRS is also used to write appropriate system verification plans. If the system contains hardware and software, then the Hardware Test Plan and Software Test Plan are also generated from the system requirements.
c) During the Implementation phase, test procedures will be defined from the SyRS.
d) During the validation process, validation procedures based on the SyRS are used to provide the customer a basis for acceptance of the system.

If any changes to the SyRS baseline are to be made, they should be controlled through a formal change management process. This process should include appropriate negotiation among parties affected by the change and should trigger pertinent risk assessments (e.g., schedules or costs).

4.5 Benefits

A properly written SyRS benefits all subsequent phases of the life cycle in several different ways. The SyRS documents the complete set of system capabilities and provides the following benefits:

a) Assurance to the customer that the technical community understands customer's needs and is responsive to them
b) An early opportunity for bi-directional feedback between the customer and the technical community.
c) A method for the customer and the technical community to identify problems and misunderstandings while relatively inexpensive to correct
d) A basis for system qualification to establish that the system meets the customers' needs
e) Protection for the technical community, providing a baseline for system capabilities and a basis of determining when the construction of the system is complete
f) Support for the developer's program planning, design, and development efforts
g) Aids in assessing the effects of the inevitable requirement changes
h) Increased protection against customer and technical community misunderstandings as development progresses

4.6 Dynamics of system requirements

Requirements are rarely static. Although it is desirable to freeze the set of requirements permanently, it is rarely possible. Requirements that are likely to evolve should be identified and communicated to both customers and technical community. A core subset of requirements may be frozen early. The impact of purposed new requirements must be evaluated to ensure the initial intent of the requirements baseline are maintained or changes to the intent are understood and accepted by the customer.

5. SyRS development process overview

This clause provides an overview for the steps in the SyRS development process. The system requirements development process, in general, interfaces with three external agents—the customer, the environment, and the technical community. Each of the external agents are described in the text below. Figure 1 shows the interactions among the various agents necessary to develop an SyRS.

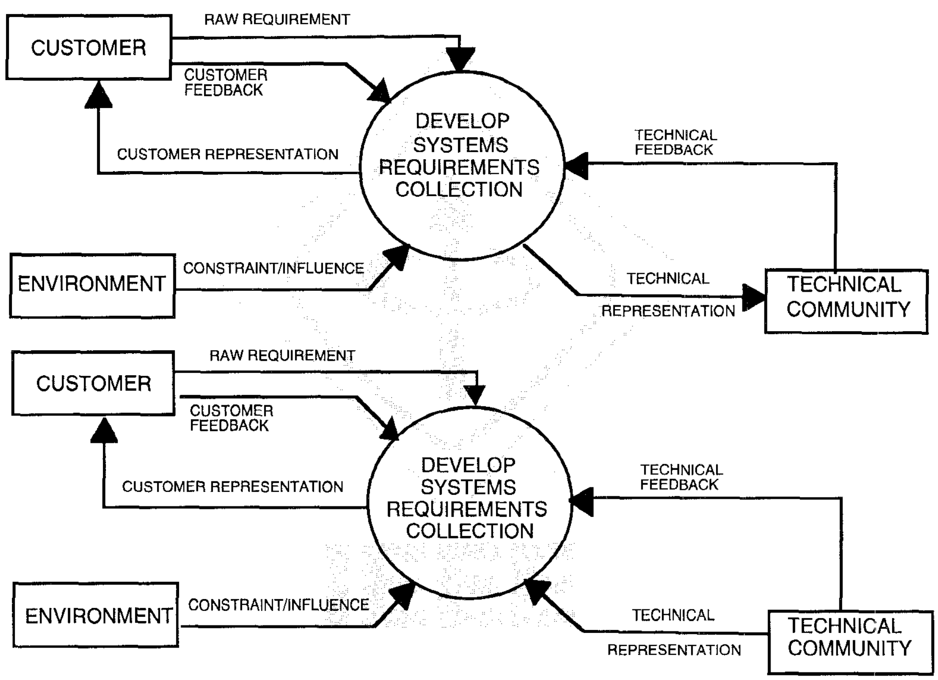

Figure 1—Context for developing an SyRS

5.1 Customer

Customers are the keystone element of the SyRS context. They are prime system drivers providing their objectives, needs, or problems to the SyRS process. The exchange between customers and SyRS developers are discussed below.

5.1.1 Raw requirements

Prior to the SyRS process the customer has an idea for a system, for a process improvement, or a problem to be solved. At this point, any initial concept for a system may be imprecise and unstructured. Requirements will often be intermingled with ideas and suggestions for potential designs. These raw requirements are often expressed in initiating documents similar to the following:

a) Concept of operations: This type of document focuses on the goals, objectives and general desired capabilities of the potential system without indicating how the system will be implemented to actually achieve the goals.
b) System concept: This type of document includes concept of operations information, but will also include a preliminary interface design for the system and other explicit requirements.
c) Marketing specification: This type of document includes a features list (often in bullet format) for a new system or systems and will identify the scope of the features and their priority (which are mandatory or highly desirable) to provide an edge in the marketplace. It also includes a context or boundary defining how the new system must interact with existing systems. A cost/benefit analysis, and required delivery schedule may be provided.
d) Request for proposal: In some instances a Request for Proposal will be prepared. This may include one or more of the above initiating documents. Its purpose will be to solicit bids for consideration from several sources to construct a system or may simply require assistance to generate system initiating documents.
e) External system interfaces: The definition of all interfaces external to the system, literally or by reference, is one of the most important (yet most often overlooked) activities to be accomplished prior to the generation of the SyRS. An approved definition of the system's external universe reasonably bounds or restricts what the system is required to do internally. All known elements of each separately defined interfaces should be described. This information may be included in the SyRS if it is not too voluminous. However, in most cases it is better to have a System External Interface Control Document (ICD). There are many types of possible interfaces external to the system and a single system may have several interfaces of differing types. The following list provides some examples:
   - operational
   - computer to computer
   - electrical
   - data links and protocols
   - telecommunications lines
   - device to system, system to device
   - computer to system, system to computer
   - environmental sense and control interfaces

5.1.2 Customer representation

Feedback to the customer includes SyRS representations and technical interchange or communications clarifying and/or confirming requirements.

5.1.3 Customer feedback

Customer feedback includes information updating the customers objectives, problems, or needs, and modifying requirements concerning technical interchange communications, and identifying new requirements.

5.2 Environment

In addition to the analyst and the customer, the environment can implicitly or explicitly influence or place a constraint the on system requirements. The analyst should be aware of these influences on system capabilities. In cases where systems are sensitive to environmental influences, the customers and analyst will specify environmental influences that effect system requirements. Environmental influences that affect system requirements are described. Environmental influences can be classified into overlapping categories, as follows:

a) Political
b) Market
c) Standards and technical policies
d) Cultural
e) Organizational
f) Physical

These categories are described below. However, it must be realized that these descriptions are represent factors that should be considered, but this list is not to be construed as exhaustive.

5.2.1 Political influence

International, federal, state, and local governmental agencies have laws and regulations that influence system requirements. Some governmental agencies may have enforcement organizations that check for compliance with their laws and regulations. Examples of governmental laws are copyright, patent, and trademark laws. Examples of governmental regulations are zoning, environmental hazards, waste, recycling, system safety, and health.

Political influence changes as a function of political boundaries. What affects system requirements in one environment may be completely different in another. Therefore, it is important to conduct research in the political environment where the system will be manufactured and/or used to ensure that the system conforms to all of the governmental laws and regulations.

5.2.2 Market influence

There are three types of market conditions influence the development of the systems specification. The first matches the customer's needs to the systems using marketing research or by developing markets to match technical research. Matching the customer's needs to systems affects the system requirements up front and becomes part of the customer requirements.

The second market influence is demand fulfilling influence. This influence is part of the environmental inputs as shown in figure 1. The demand fulfilling influence must be considered because it affects system distribution and accessibility. Distribution and accessibility adds to the system requirements. For example, the system should be lightweight to reduce shipping cost, or the system must be of small size to fit into vending machines. Distribution and accessibility requirements must be identified during system development and before the system is manufactured and/or integrated to allow these requirements to be incorporated into the system. Without easy access to the system, the system success will be limited. Therefore, it is important to consider the market segments for which the system is targeted and to consider how marketing information can be used to derive system requirements.

The third market influence is competition. Knowing a competitor's systems will help define requirements. To stay competitive, the following should be considered:

a) Functionality
b) Price
c) Reliability
d) Durability
e) Performance
f) Maintainability
g) System safety & security

Analyzing the competitive market is a continuous process and will affect requirements for both new and existing systems. Systems can evolve into completely new systems that may have little resemblance to the customers original concepts.

5.2.3 Standards and technical policies influence

System requirements are influenced directly by customers who have to conform to standards and technical policies issued by government or industries. Technical policies and associated standards and guidelines help ensure the following:

a) System consistency
b) System safety
c) System reliability and maintainability

System consistency standards and guidelines prescribe specific requirements by providing details of how a particular system should be implemented.

Industrial safety standards are generally imposed to help prevent safety hazards and potential legal problems. Safety compliance requirements should be clearly identified in a system requirements document (i.e., safety requirements for toy manufacturing industry, UL certifications, National Electrical Code requirements)

The customer and technical community may require that the system should pass certain reliability criteria as prescribed in technical standards. Reliability and maintainability requirements should be identified in the SyRS. These requirements can come in many forms. For instance, they may be directly system oriented and may require specifications of maximum outage time and minimum mean time between failure, or minimum mean time to repair.

5.2.4 Cultural influence

Culture is the integrated human behavior patterns that are transmitted from generation to generation. It is a learned experience that originates from religious beliefs, country of origin, ethnic groups, socioeconomic levels, language, media, place of employment, and immediate family. To understand the culture of a region or market segment, the values and beliefs of the people must be known. Cultural influence should be considered when developing a system because it will affect the system requirements.

5.2.5 Organizational influence

System requirements are influenced by the organization in which the requirements are developed. Company organizational influence can take the form of marketing, internal politics, technical policies, and internal standards. The company's influences on system requirements are similar to those of the other spheres of influence plus its own unique set; that is, every company has its own culture, purposes, values and goals that can and will influence the system that it develops, manufactures, and/or delivers.

5.2.6 Physical influence

The physical influence includes natural and man-made influences such as temperature, radiation, moisture, pressure, and chemical.

5.3 Technical community

The technical community represented in figure 1, is composed of those involved in the activities of system design, implementation, integration, test, manufacturing, deployment, operations, and maintenance. All elements of the technical community should be involved in the SyRS development process as early as possible. Early inclusion of the technical community provides a mechanism for the SyRS developers to reduce the possibility that new requirements or changes to the original requirements may be discovered later in the system life cycle.

5.3.1 Technical representation

Representations of the requirement collection, prepared for the technical community, may include technical interchange or communications that clarify and/or confirm requirements.

5.3.2 Technical feedback

The technical community provides feedback during various activities that can cause modification, additions and/or deletions to the requirement collection. The SyRS is refined as necessary to support subsequent life cycle phases of the system.

For example, following the requirement phase, a system test plan is developed where individual requirements are allocated to specific tests. This process can reveal requirements that are non-testable resulting in modification of the SyRS to ensure testability.

Other feedback from the technical community may provide customers with the most recent system features, upcoming technologies and insight into advanced implementation methods.

6. Well-formed requirements

This clause explains the properties of a well formed requirement, provides an example of a well-formed requirement, and points out requirement pitfalls.

6.1 Definition of well-formed requirement

As previously defined, a well-formed requirement is a statement of system functionality (a capability) that can be validated, must be met or possessed by a system to solve a customer problem or to achieve a customer objective, and is qualified by measurable conditions and bounded by constraints.

This definition helps in the classification of general customer requirements. Requirements can be taken from customer needs and can be derived from technical analysis. The definition provides a means for distinguishing between requirements as capabilities and the attributes of those requirements (conditions and constraints). Constraints may be functional or nonfunctional. An example of a nonfunctional constraint might be that the system be painted a particular shade of blue solely for non-required decorative purposes.

This guide recommends that system implementation process requirements, such as mandating a particular design methodology, not be included in an SyRS. Process requirements should be captured in other system controlling technical documentation such as quality plans, contracts, or statements of work.

6.1.1 Capabilities

Capabilities are the fundamental requirements of the system and represent the features or functions of the system needed or desired by the customer. A capability should usually be stated in such a way that it describes what the system should do. The capability should also be stated in a manner which is solution independent. This will permit consideration of different ways of meeting the need or of providing the feature or function. For example, capabilities of a high-speed rail system between Los Angeles and New York could include the ability to start, to accelerate, to cruise, to decelerate, to stop, to load passengers, to unload passengers. A capability, the high-speed rail system, does not include the brand of the computer operating system.

6.1.2 Conditions

Conditions are measurable qualitative or quantitative attributes and characteristics that are stipulated for a capability. They further qualify a capability that is needed, and provide attributes which permit a capability to be formulated and stated in a manner that can be validated and verified. For example, in the high-speed rail system mentioned above, a condition of the capability to cruise, may be a cruise range of 0 to 300 km/hr or an optimal cruise rate of 200 km/hr.

It makes sense to include conditions (measurable attributes) only if they apply to something to be measured such as a capability. For example, it is meaningless to have a requirement of a system that states 0 to 200 km/hr in the abstract. This range can qualify a cruising speed for a high-speed rail link but not the speed at which an elevator should lift passengers.

Conditions may limit the options open to a designer. It is important to identify conditions as attributes of capabilities, not as primary capabilities, to ensure that the requirements clearly define the need without imposing unnecessary bounds on the solution space.

6.1.3 Constraints

Constraints are requirements that are imposed on the solution by circumstance, force, or compulsion. Constraints limit absolutely the options open to a designer of a solution by imposing immovable boundaries and limits. For example, the high-speed rail link mentioned above will be constrained by the need to get people to their destination alive (a safety constraint could be mandatory seat-belts) and could be constrained by technology (the customer may require that all train control software be written in Ada).

A list of constraints can include interfaces to already existing systems (e.g., format, protocol, or content) where the interface cannot be changed, physical size limitations (e.g., a controller must fit within a limited space in an airplane wing), laws of nature, laws of a particular country, available time or budget, priority (e.g., mandatory or optional), or a pre-existing technology platform.

Constraints may apply across all requirements or be specified in a relationship to a specific capability or set of capabilities.

Constraints may be identified as stand-alone requirements, that is, not bounding any specific capability, or as constraints upon individual capabilities. Many constraints, such as the choice of technology (e.g., the type of operating system), will apply to the entire set of capabilities. Other constraints will apply to only a single or a few capabilities. For example, there will be safety constraints imposed on acceleration for a high-speed rail system that will not apply to the passenger loading functions.

6.1.4 Example

The purpose of this example is to state a well-formed requirement and its associated conditional and capability requirements, as follows:

Move people from New York to California at a maximum speed of 5300 km/hr.

> capability: move people between Los Angeles and New York
> condition: cruising speed of 2500 km/hr
> constraint: maximum speed of 300 km/hr

6.2 Properties of a requirement

Each requirement should possess the following properties:

a) Abstract: Each requirement should be implementation independent
b) Unambiguous: Each requirement should be stated in such a way so that it can be interpreted in only one way
c) Traceable: For each requirement it should be feasible to determine a relationship between specific documented customer statement(s) of need and the specific statements in the definition of the system given in the SyRS as evidence of the source of a requirement
d) Validatable: Each requirement should have the means to prove that the system satisfies the requirements

6.3 Categorization

To support the analysis of requirements, the requirements should be categorized by their identification, priority, criticality, feasibility, risk, source, and type. Each of these categories is described in more detail below:

a) Identification: Each requirement should be uniquely identified (i.e., number, name tag, mnemonic, buttons, hypertext). Identification can reflect linkages and relationships, if needed, or they can be separate from identification.
b) Priority: The customer should identify the priority of each requirement. This may be established through a consensus process among potential customer. As appropriate, a scale such as 1:10 or a simple scheme such as High, Medium, Low, Out, could be used for identifying the priority of each requirement.
c) Criticality: The analyst, working with the customer, should define the criticality of each requirement. Some requirements could have a low priority from the users perspective, but nevertheless be essential for the success of the system. For example, a requirement to measure external ambient temperature could be essential to provide support to other requirements such as the maintenance of internal cabin temperature. This relationship should be identified so if the primary requirement is removed by the customer, the supporting requirement can also be eliminated.
d) Feasibility: The customer and analyst working together should identify the feasibility of including each particular requirement in the system, and classify each requirement by types of feasibility appropriate to the system domain. Feasibility could be based upon an understanding of such things as the current state of technology (e.g., commercially available components vs. original research), the customer's environment (e.g., readiness or capability to accept change), and the risk or cost associated with a particular requirement.
e) Risk: Risk analysis techniques can be used to determine a grading for system requirements. In terms of their consequences or degree of risk avoidance, major risks are related to potential financial loss, environmental impact, safety and health issues, and national standards or laws.
f) Source: Each requirement should be further classified by a label that indicates the originator. There may be multiple sources that can all be considered creators of the requirement. It is useful to identify the creator(s) of each requirement so that if requirements are unclear, conflict, or need to be modified or deleted it will be possible to identify the individual(s) or organization(s) to be consulted.
g) Type: Requirements can also be categorized by one or more of the following types:

—input (e.g., receive EDI data)
—output (e.g., export a particular format)
—reliability (e.g., mean time to failure)
—availability (e.g., expected hours of operation)
—maintainability (e.g., ease with which components can be replaced)
—performance (e.g., response time)

—accessibility (e.g., different navigation paths for novice and experienced users)
—environmental conditions (e.g., dust levels that must be tolerated)
—ergonomic (e.g., use of particular colors to reduce eye strain)
—safety (e.g., below specified limits for electrical magnetic radiation)
—security (e.g., limits to physical, functional or data access, by authorized or unauthorized users)
—facility requirements (e.g., use of domestic electrical current)
—transportability (e.g., weight limits for portability)
—training (e.g., includes tutorials or computer-based training)
—documentation (e.g., on-line help facility)
—external interfaces (e.g., support for industry standard communication mode/format)
—testing (e.g., support for remote diagnostics)
—quality provisions (e.g., minimum required calibration intervals)
—policy and regulatory (e.g., environmental protection agency policies)
—compatibility to existing systems (e.g., uses analog telephone system as default mode)
—standards and technical policies (e.g., products to conforms to ASME codes)
—conversion (e.g., will accept data produced by older versions of system)
—growth capacity (e.g., will support an additional number of users)
—installation (e.g., ability to put a new system into service)

6.4 Pitfalls

Some pitfalls to avoid when building a well-formed requirements are the following:

a) Design & Implementation: There is a tendency on the part of analysts and customers who are defining requirements to include design and implementation decisions along with the requirements statements. Such information may still be important. In this case, the information should be documented and communicated in some other form of documentation in order to aid in design and implementation.
b) Overspecification:
   —requirements that express an exact commercial system set or a system that can be bought rather than made (these are not an expression of what the system should do)
   —requirements that state tolerances for items deep within the conceptual system (frequently stated as error requirements at very low levels)
   —requirements that implement solutions (Requirements state a need.)
c) Overconstrained: Requirements with unnecessary constraints (For example, if a system must be able to run on rechargeable batteries, a derived requirement might be that the time to recharge should be less than 3 h. If this time is too restrictive and a 12-h recharge time is sufficient, potential solutions are eliminated.)
d) Unbounded:
   —requirements making relative statements (These requirements cannot be verified and may only need to be restated. For example, the requirement to "minimize noise" may be restated as "noise levels should not exceed...")
   —requirements that are open-ended (frequently stated as "including, but not limited to..." or lists ending in "etc.")
   —requirements making subjective or vague statements (frequently contain terms as user friendly, quick response time or cost effective)
e) Assumptions:
   —requirements based on undocumented assumptions (The assumption should be documented as well as the requirement.)
   —requirements based on the assumption that a particular standard or system undergoing development will reach completion (The assumption and an alternative requirement should be documented.)

7. SyRS development

System requirements development, shown in figure 1, is an iterative process. The four subprocesses are shown in figure 2. These subprocesses are as follows:

a) Identify requirements from the customer, the environment, and the experience of the technical community
b) Build well-formed requirements
c) Organize the requirements into an SyRS
d) Present the SyRS in various representations for different audiences

The purpose of decomposing the system requirements development process into subprocesses is to aid in the full and accurate development of the SyRS. The subprocesses below are presented as occurring sequentially. However, there will often be a degree of subprocess overlap or iteration.

The iterative application of this process results in the ongoing modification of the SyRS. Usually modifications are applied against a The SyRS baseline, and managed under change control procedures. See IEEE Std 1220-1994 for change control procedures.

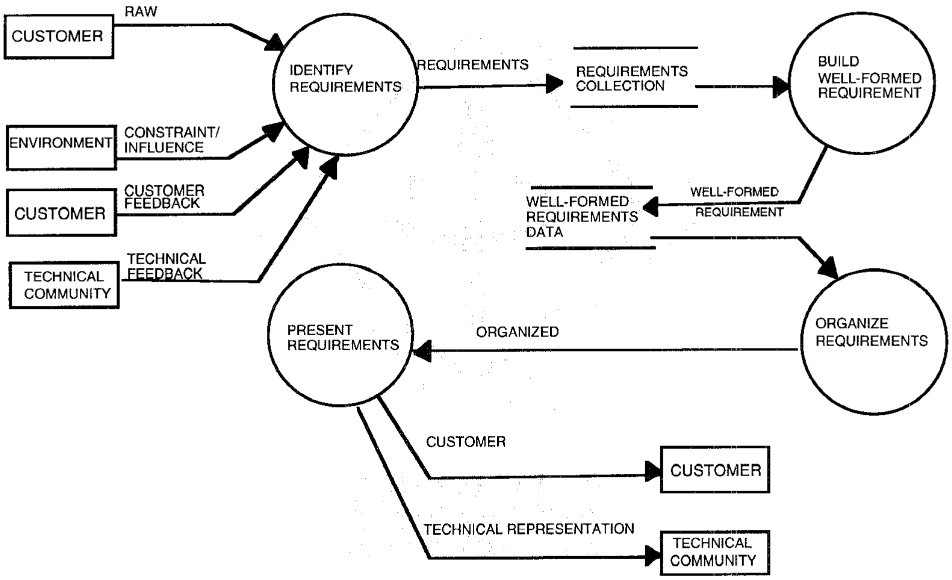

Figure 2—SyRS development process

7.1 Identify requirements

Working with customers, analysts filter the various inputs identified in figure 2 and extract a set of requirements, establish necessary derived requirements, and create requirements. Requirements may be extracted from initiating documents and through analytical exercises conducted with customer. The goal of this iterative process is to solicit all system requirements, to ensure each requirement is stated once, and to ensure that none are missed.

There are a variety of approaches that can be followed in the identification of customer requirements. In practice, each organization will have its own approach to identifying requirements and initiating the process of creating a system solution. In some organizations, customers will undertake the entire process in-house. In this case the identification and specification of requirements will be driven by the customer. In other organizations, customers will identify a preliminary set of requirements and will request assistance from an analyst within their organization or through a contract with an external consultant or systems integrator.

The analyst, whether from within the organization or external, will work with the customer to identify and structure the requirements. In some organizations the analyst will work directly with customer. In other organizations, the analyst will not be given direct access to the customer and will have to work through one or more layers of intermediaries (technical, legal, administrative) who represent the customer.

Because of the dynamics involved in identifying requirements, it is important that the customer and analyst agree on the process. The analyst needs to prepare a plan to guide the process, define representations of the SyRS that will be produced for different audiences, and identify the tools and techniques that will be used.

The process should move forward toward the goal of developing the SyRS database and provide customers with a system that will meet their needs. The analyst should ensure that the identification of requirements uses all appropriate techniques within this process.

A requirements management process should be used to ensure the following:

a) The process is goal-directed and aimed at the production of a set of requirements
b) The system boundaries are defined
c) All requirements are solicited, fairly evaluated, and documented
d) Requirements are specified as capabilities and that qualifying conditions and bounding constraints are identified distinctly from capabilities
e) Requirements are validated, or purged if invalid, from the requirements set
f) Consideration is given to consistency when many individuals ("authors") may be contributing to the development of the requirements set
g) The developing requirements set is understood, at an appropriate level of detail, by all individuals participating in the process

7.1.1 Techniques for identifying requirements

Requirements begin as ideas or concepts that may originate from a response to a perceived threat to security or market share, from an imposition of legislation or regulation, from a desire to create a new or better system or process, from the need to replace an existing system, or from some other actual or perceived need. Although the ideas or concepts may originate with one individual, a set of requirements usually is best obtained from the interaction of a group where ideas are shared and shaped.

There are a number of techniques for identifying requirements. These include the following:

a) Structured workshops
b) Brainstorming or problem-solving sessions
c) Interviews
d) Surveys/questionnaires
e) Observation of work patterns (e.g., time and motion studies)
f) Observation of the system's organizational and political environment (e.g., sociogram)
g) Technical documentation review
h) Market analysis
i) Competitive system assessment
j) Reverse engineering
k) Simulations
l) Prototyping
m) Benchmarking processes and systems

7.1.2 Interaction between customers and analysts

In a situation where an analyst has been hired to work with a customer, it will be necessary to establish an effective process of interaction between the two parties. To make this interaction effective, each party needs to understand that they have a role in educating the other party and that they should work together to jointly define the requirements.

7.1.2.1 Mutual education

Education must be a two-way process. First, the analysts need to learn about the customer's environment, current systems (if any), and requirements. Time and effort need to be allocated on both sides to accommodate this education process.

Second, customers may also need education. They may need education from the analysts during the process of identifying and specifying requirements. In addition, the analysts may be needed to educate customers with respect to the requirements themselves and contribute requirements from their experience.

7.1.2.2 Joint definition of requirements

There are multiple ways in which the customer and analyst may interact in the process of defining requirements. For example, analysts may simply conduct interviews to solicit input and then organize and present the requirements for review by the customer.

The analysts' experience is very important, but it should not bias or stifle the customer's involvement in any way. As they work with the customer, their primary objective should be to solicit and organize requirements derived from the customer. They should add requirements from their own experience or from previously defined system solutions only when these additional requirements have been missed by the customer, clearly add value for the customer to the system being specified, and have been approved by the customer.

As another example, personnel from the customer may be involved in workshop sessions with the analyst. At these workshops there can be a significant amount of brainstorming and interactive definition of requirements. These sessions are usually driven and directed by the analyst. The results of the workshop are documented by the analyst.

A more cooperative (joint definition) approach may involve the customer even more directly in the definition of requirements. Customer personnel may participate in the definition of the requirements to the extent that they also author the SyRS.

Whatever techniques are used, the objective is to define the requirements while building consensus and a common level of understanding. The customer and the analyst need to come to a point where they have a common understanding of the requirements and can represent these consistently as an SyRS, to the satisfaction of the customer.

7.2 Build a well-formed requirement

The analysts carry out this subphase by doing the following:

a) Ensuring that each requirement is a necessary, short, definitive statement of need (capability, constraints)
b) Defining the appropriate conditions (quantitative or qualitative measures) for each requirement and avoiding adjectives like resistant or industry wide
c) Avoiding requirements pitfalls (see 6.4)
d) Ensuring the readability of requirements, which entails the following:

—simple words/phrases/concepts
—uniform arrangement and relationship
—definition of unique words, symbols and notations
—the use of language and symbology should be grammatically correct

e) Ensuring testability

The following is an example of how to generate a well-formed requirement from an initial customer statement. The customer statement "Move people between Los Angeles and New York" is a statement of a raw requirement and is the basis of a well-formed requirement. Further information from the customer includes, "speed should be no greater than 300 km/hr and cruising speed should be 200 km/hr." These conditions and constraints can be applied to the following raw requirement.

capability: move people between Los Angeles and New York
condition: cruising speed of 200 km/hr
constraint: maximum speed of 300 km/hr
well-formed requirement: this system should move people between Los Angeles and New York at an optimal cruising speed of 200 km/hr with a maximum speed of 300 km/hr.

With a single capability, there can be multiple conditions and constraints. However, additional conditions and/or additional constraints on a capability may necessitate definition of additional well-formed requirements.

7.3 Organize requirements

In this subprocess, analysts add structure to the set of requirements by relating the requirements to one another according to some comparative definition method. Certain tasks in this process can be supported by automation.

This activity is characterized by the following:

a) Searching for patterns around which to group requirements
b) Utilizing experience and judgment to account for appropriate technical approaches
c) Utilizing creativity and intuition to generate alternative approaches and to prioritize requirements based on customer inputs
d) Defining the requirements properties
e) Defining the requirement attributes

Requirement attributes can be assigned to each well-formed requirement. For example:

<identification> = 2.1.3.6
<priority> = high
<criticality> = low
<feasibility> = high
<risk> = medium
<source> = customer
<type> = performance

There are various schemes for organizing specifications into an ordered set. The most often used scheme is to assemble requirements into a hierarchy of capabilities where more general capabilities are decomposed into subordinate requirements. Another scheme is to use network links (e.g., hypertext) shows the relationship between all the lowest level requirements. Whatever scheme is used, the SyRS should indicate the relationships among the requirements. The specific relationships will depend on the methods, techniques, and tools used to capture, record, and store the requirements. Some of the relationships among the requirements which can be maintained in an SyRS include the following hierarchical dependencies:

a) Events
b) Information/data
c) Physical or abstract objects
d) Functions

7.4 Present requirements

In this subprocess, analysts working with the customer identify the best means of communicating the requirements to all individuals who need to understand, review, accept, or use the SyRS. A single representation is not always suitable because:

a) The customer and technical community usually have different cultures and languages, thus, the same system requirements may have to be presented differently to the technical or customer communities
b) Retrieval of specific information is difficult in some representations
c) Representation of interactions can be difficult to do in some representations
d) Relating information in one place to information in another place can be difficult in some representations

Therefore, it is important that the analysts, working with the customer, identify the best means of communicating the requirements to all individuals who need to understand, review, accept, or use the SyRS. To accomplish this, different representations should be prepared from the SyRS. These representations should not be separately maintained, rather they should be representations derived from, and generated from the SyRS. For example an overview document may be produced for customer management that contains annotation narrative and a selected subset of the high-level requirements. For the individual who is responsible for accepting the requirements on behalf of the customer, a more detailed document, including formal models, may be generated. For the design team, a complete set of formal models may be generated. Automated tools could be used productively both to maintain the SyRS and to generate different representations.

7.4.1 Methods of representation

Methods of representation may be one or a combination of the following so that they allow appropriate views of the requirements depending on the needs of the audience:

a) Textual
   — paper
   — electronic
b) Model
   —physical
   —symbolic
   —graphical
   —prototype

Writing requirements generally continues beyond the approval of the SyRS. In large and complicated systems, there is a high probability that the first approved version of SyRS will contain errors of omission or distortion. In addition, many systems will evolve with the additions of new features. This necessitates the repeat or iteration of this process to correct the initial requirements errors or add new requirements to enhance the system's features. The formal control of SyRS is critical to manage the changes to the SyRS.

Annex A

(informative)

An SyRS outline

This guide recognizes and endorses a wide variety of techniques and media to communicate requirements including text and models. The purpose of the following is to help focus on the technical content of an SyRS. See IEEE Std 1220-1994 for process requirements for developing an SyRS. The representation and content can be expanded or contracted for the customer or technical community. There are many possible representations of an SyRS and the following is merely one example:

An SyRS Outline

Table of contents
List of figures
List of tables

1. Introduction

1.1 System purpose
1.2 System scope
1.3 Definitions, acronyms, and abbreviations
1.4 References
1.5 System overview

2. General System Description

2.1 System context
2.2 System modes and states
2.3 Major system capabilities
2.4 Major system conditions
2.5 Major system constraints
2.6 User characteristics
2.7 Assumptions and dependencies
2.8 Operational scenarios

3. System capabilities, conditions, and constraints

Note—System behavior, exception handling, manufacturability, and deployment should be covered under each capability, condition, and constraint.

3.1 Physical
3.1.1 Construction
3.1.2 Durability
3.1.3 Adaptability
3.1.4 Environmental conditions
3.2 System performance characteristics
3.3 System security
3.4 Information management
3.5 System operations
3.5.1 System human factors
3.5.2 System maintainability
3.5.3 System reliability
3.6 Policy and regulation
3.7 System life cycle sustainment

4. System interfaces

Outline explanation

For those items that are not self-explanatory, the following information is provided:

1.2 System scope

This subclause should

a) Identify the system to be produced by name.
b) Refer to and state the results of the earlier finalized needs analysis, in the form of a brief but formal expression of the customer's problem(s). Explain what the system will and will not do to satisfy those needs.
c) Describe the application of the system being specified. As a portion of this, it should describe all relevant top-level benefits, objectives, and goals as precisely as possible.

2.1 System context

Appropriate diagrams and narrative should be included in this subclause, to provide an overview of the context of the system, defining all significant interfaces crossing the system's boundaries.

2.2 System modes and states

If the system can exist in various modes or states, this subclause should describe these, and as appropriate, using diagrams.

2.3 Major system capabilities

This subclause should provide diagrams and accompanying narrative to show major capability groupings of the requirements.

2.6 User characteristics

This subclause should identify each type of user of the system (by function, location, type of device), the number in each group, and the nature of their use of the system.

2.8 Operational scenarios

This subclause should provide descriptive examples of how the system will be used.

3 System capabilities, conditions, and constraints

3.1 Physical

3.1.1 Construction

The environmental (mechanical, electrical, chemical) characteristics of where the system will be installed should be included. For examples, the weight limits of the system, moments of inertia, dimensional and volume limitations, crew space, operator station layout, ingress, egress, and access for maintenance should be specified here. Requirements for materials to be used in the item or service covered by this specification should be stated. Requirements covering name-plates and system markings, interchangeability of equipment, and workmanship should be covered in this subclause.

3.1.3 Adaptability

Growth, expansion, capability, and contraction should be included in this subclause. For example, if the system will require future network bandwidth, the hardware rack should be specified with extra slots to accommodate new network cards as demand increases.

3.1.4 Environmental conditions

This subclause should include environmental conditions to be encountered by the system. The following subjects should be considered for coverage: natural environment (wind, rain, temperature); induced environment (motion, shock, noise); electromagnetic signal environment.

3.2 System performance characteristics

This paragraph should be used to highlight the critical performance conditions and their associated capabilities. As a general guide, include such considerations as:

a) Dynamic actions or changes that occur (e.g., rates, velocities, movements, and noise levels).
b) Quantitative criteria covering endurance capabilities of the equipment required to meet the user needs under stipulated environmental and other conditions, including minimum total life expectancy. Indicate required operational session duration and planned utilization rate.
c) Performance requirements for the operational phases and modes.

3.3 System security

System security requirements related to both the facility that houses the system and operational security requirements should be given in this paragraph. One example of security requirements might be to specify the security and privacy requirements, including access limitations to the system, such as existence of logon procedures and passwords, and of data protection and recovery methods. This could include the factors that would protect the system from accidental or malicious access, use, modification, destruction, or disclosure. Especially in safety-critical embedded systems this should incorporate a distributed log or history of data sets, the assignment of certain functions to different single systems, or the restriction of communications between some areas of the system.

3.5.1 System human factors

This subparagraph should reference applicable documents and specify any special or unique requirements, e.g., constraints on allocation of functions to personnel and communications and personnel/equipment interactions. Included should be those specific areas, stations, or equipment that would require concentrated human engineering attention due to the sensitivity of the operation or criticality of the task (i.e., those areas where the effects of human error would be particularly serious).

3.5.2 System maintainability

This subclause should specify the quantitative maintainability requirements. The requirements should apply to maintenance in the planned maintenance and support environment and should be stated in quantitative terms. Examples are as follows:

a) Time (e.g., mean and maximum downtime, reaction time, turnaround time, mean and maximum times to repair, mean time between maintenance actions)
b) Rate (e.g., maintenance staff hours per specific maintenance action, operational ready rate, maintenance hours per operating hour, frequency of preventative maintenance)
c) Maintenance complexity (e.g., number of people and skill levels, variety of support equipment)
d) Maintenance action indices (e.g., maintenance costs per operating hour, staff hours per overhaul)

3.5.3 System reliability

The following subclauses should specify the system reliability requirements in quantitative terms, and should define the conditions under which the reliability requirements are to be met. The system reliability subparagraph may include the reliability apportionment model to support allocation of reliability values assigned to system functions for their share in achieving desired system reliability.

3.6 Policy and regulation

This subclause should detail any relevant organizational policies that will affect the operation or performance of the system as well as any relevant external regulatory requirements, or constraints imposed by normal business practices. Examples of requirements include multilingual support, labor policies, protection of personal information and reports to a regulatory agency.

Health and safety criteria, including those basic to the design of the system, with respect to equipment characteristics, methods of operation, and environmental influences should be specified in this subclause.

Requirements covering toxic systems and electromagnetic radiation should be covered in this subclause.

3.7 System life cycle sustainment

This subclause should outline quality activities, such as review, and measurement collection and analysis, to help realize a quality system.

4 System interfaces

This clause contains the specification of requirements for interfaces among different components and their external capabilities, including all its users, both human and other systems. The characteristics of interfaces to systems under development, or future systems, should also be included. Any interdependencies or constraints associated with the interfaces should also be identified (e.g., communication protocols, special devices, standards, fixed formats). Each interface may represent a bi-directional flow of information. A graphic representation of the interfaces should be used when appropriate for the sake of clarity.

NOTE—This outline does not include all capability categories pertinent to all fields. For example, it does not cover communications, storage, distribution, sensors, instrumentation. Also, these above clauses can be organized in any way the reader sees fit.

Annex B

(informative)

Bibliography

[B1] Blanchard, Benjamin S. *System Engineering Management.* Wiley-Interscience, 1991.

[B2] Blanchard, Benjamin S., and Walter J. Fabrycsky. *Systems Engineering & Analysis.* International Series and Industrial & Systems Engineering, Prentice Hall, 1990.

[B3] Gause, Donald C., and Gerald M. Weinberg. *Exploring Requirements: Quality Before Design.* Dorset House Publishing, New York, 1989.